User's Guide WAQUA: General Information

Rijkswaterstaat
Ministerie van Infrastructuur en Milieu

User's Guide WAQUA: General Information

Version number : Version 10.59, October 2013
Maintenance : see www.helpdeskwater.nl/waqua

Copyright : Rijkswaterstaat

Log-sheet

Table 1: Log-Sheet

Document version	Date	Change no	Changes with respect to previous version
10.27	13-10-2003	P03022 P03023 P03031 P03032	condition on wind direction / speed not allowed at SVWP. waqpre-input description for barriers improved. improvements for slib3d. layout improvements (vector arrows, integral signs, etc.)
10.28	03-11-2003	ruwh02-b	Introduction of roughness combinations and ebb/flood roughnesses
10.29	10-11-2003	dynbar02	extension of dynamic barrier steering
10.30	14-11-2003	P03047	general check (export 2003-02)
10.31	16-02-2004	P04003	removal of (sub-)systems: waqua-oud, sispos, conidp, sds-kom, sdsmap, sdspri, sdsres, sds11b
10.32	14-06-2004	DDHV01	combined horizontal and vertical domain decomposition
10.33	28-06-2004	P03043	nu-signs made more clear in Navier-Stokes equation
10.34	05-07-2004	P04008	installation of RUWH03.
10.35	13-07-2004	kalwnd01	Introduction of multiple formats in waqwnd
10.36	01-09-2004	P03033 P04009	Description of approved weir-routines Update QAD-boundary conditions (VORtech)
10.37	25-03-2005	DROOGV01 M04042 M04069 P04001 P04012 P05002 P05003 P05005 P05006 P05008 W04006 W04007 W04008 W04013	phase 2: new drying/flooding functionality use defined distance (in routine "wasrv1") correction in routine "wapttd" extra functionality in simpar correction in slib3d (from w04007, decay) corrections in waqpre for droogv01 (phase 2) collect class values (new subsystem "waqclv") correction in check on "tlsmooth" at restart mark zero-differences in Simona-boxes (waqview) improved barrier model (waqpro) add maps for dynamic barriers introduce decay in slib calculations (slib3d) save 3D-vertical eddy diffusion coefficients on sds-file extension of w04008 (in slib3d)
10.38	21-07-2005	DROOGV01	phase 3 – new drying/flooding functionality

Document version	Date	Change no	Changes with respect to previous version
10.39	03-08-2005	B05007 M04036 P05014 P05016 W04003 W05006	updates for "class values" triwaq-correction in routine "wascht" correction in routine "waswcv" ("class values") extra checks for TIME_AND_VALUES incorporation of User's Guide couple roughness-codes extended: 1201-1300 to 1201-1400
10.40	31-08-2005	M05069	introduction of GSC
10.41	01-11-2005	M05045 M05087 M05091 W05001 W05007	backtracking in simpar correction in coppre / coppos extra conditions for barrier steering reference to FAQ for template introduction of "minvalues"
10.42	20-06-2007	C71236	keyword CDCON, ITERACCURACY and application WAQRIV removed
10.43	28-11-2007	C71236	removed Waqbhd
10.44	22-05-2007	C66699	Introduced linear bottom friction model
10.45	20-06-2007	C71236	Removed CCO-file and other obsolete functionalities
10.46	21-06-2007	C71236	Removed program Waqriv
10.47	05-09-2007	M321608	Corrections w.r.t. missing figures and equations
10.48	19-09-2007	C74807	Changes w.r.t. calculation of energyloss for weirs
10.49	29-11-2007	C71236	Removed program Waqbhd
10.50	31-01-2008	C77132	Implementation of weirs for 3D TRIWAQ models
10.51	17-03-2008	C80842	Merged technical documentation of WAQUA and TRIWAQ: changed general introduction of weirs.
10.52	14-08-2009	C91583	Introduction of barrier-barrier structures
10.53	07-09-2009	C81107	Introduction of VILLEMONTE model for weirs
10.54	01-12-2009	M385558	Corrected cross references for barrier figures
10.55	15-07-2010	C3256	Conversion to Latex
10.56	13-09-2010	C3410	Waqpan removed; uniform layout logsheet
10.57	16-05-2012	M3754	Corrections related to conversion to Latex
10.58	01-02-2013	beheer	Removed Appendix C, systemchart
10.59	16-10-2013	3964	Remove outdated OpenDA information

Contents

1 About this manual **2**

2 About SIMONA **4**

2.1 Introduction 4

2.2 SIMONA data storage 5

2.3 The input file 5

3 About WAQUA **9**

3.1 Introduction to the system 10

3.1.1 Function of the system 10

3.1.2 Features 11

3.1.3 Modelling with WAQUA 13

3.2 Geometry 15

3.2.1 Computational grid 15

3.2.1.1 Computational grid structure 15

3.2.1.2 Basic grid principle 16

3.2.1.3 Computational grid tuning and open boundaries 18

3.2.2 Physical geometry 21

3.3 Model types 22

3.3.1 Rectilinear models 22

3.3.1.1 General 22

3.3.1.2 Mathematical description differential equations 22

3.3.2 Curvilinear models 24

3.3.2.1 General 24

3.3.2.2 Mathematical description differential equations 25

3.3.3 Spherical coordinates models 26

3.3.3.1 General 26
3.3.3.2 Mathematical description differential equations 26
3.3.4 Models using space varying wind and pressure 28
3.4 General hydrodynamics 30
3.4.1 Features 30
3.4.2 Computational methods 31
3.4.2.1 Time advancement of state variables 31
3.4.2.2 Time integration 32
3.4.2.3 Forcing functions 39
3.4.2.4 Chezy coefficient computation 44
3.4.2.5 Nikuradse roughness 47
3.4.2.6 Roughcombination roughness 52
3.4.2.7 Equation of state 59
3.4.2.8 Non-hydrostatic computations 60
3.5 Special hydrodynamical objects 61
3.5.1 Barriers and sluices 61
3.5.1.1 Features 61
3.5.1.2 Computational methods 69
3.5.2 Barrier-barrier constructions 72
3.5.2.1 Weirs 73
3.5.2.2 Features 73
3.5.2.3 Computational methods 73
3.5.2.4 Weirs in TRIWAQ 74
3.6 Drying and flooding (tidal flats) 76
3.6.1 Features 76
3.6.2 Computational methods 76
3.7 Harmonic analysis of tides 83
3.7.1 Features 83
3.7.2 Computational methods 83
3.7.2.1 Mathematical representation of the tide 83
3.7.2.2 Harmonic analysis 85
3.8 Transport of constituents 87
3.8.1 Features 87

3.8.2 Computational methods . . . 87
3.8.2.1 Time advancement of state variables . . . 88
3.8.2.2 Forcing functions . . . 90
3.9 User routines . . . 92
3.9.1 Features . . . 92
3.9.2 Mathematical description . . . 92
3.9.3 Data input/output . . . 93
3.9.4 Computation . . . 94
3.9.5 Generating procedure . . . 94
3.10 Data flow . . . 95
3.10.1 Input . . . 95
3.10.2 Output . . . 95
3.11 CPU and memory usage for computational models . . . 96
3.12 WAQUA programs . . . 100

A Examples **I**
A.1 Example user routine . . . I
A.1.1 General . . . I
A.1.2 Model description . . . I
A.1.3 Input file mussel filter . . . II
A.1.4 Subroutine WASUST . . . VI

B Definition WAQUA standard version **XI**
B.1 Standard version of May 2012 . . . XI
B.1.1 SIMONA standard version subsystems . . . XI

C Harmonic Constants **XII**

Chapter 1

About this manual

The User's Guide WAQUA contains an extensive input description of the subsystems of WAQUA.

It also describes principles and backgrounds of the WAQUA system.

WAQUA is a water movement and water quality simulation system, able to perform two-dimensional computations. A part called TRIWAQ is incorporated for three-dimensional computations. The system enables the user to simulate stationary as well as non-stationary flow patterns, transport of dissolved substances, temperature distribution and sediment transport.

WAQUA is based on SIMONA, a flexible concept for the development of modelling software.

SIMONA defines an architecture for preprocessing, memory management, data storage and post-processing. SIMONA aims for structured and controlled development of software, reducing cost for maintenance and support.

Interested users are advised to read the chapters in section 1 dealing with SIMONA. More information is available in the SIMONA programmer's guide.

This document is written for users, both model builders and model managers who make computer runs for model builders.

A general introduction to the system is given in section 1 (general information) of the user's guide WAQUA. The other sections give more detailed descriptions of the WAQUA system.

The User's Guide WAQUA consists of 8 sections:

section 1	**General Information**
	About the manual
	About SIMONA
	About the WAQUA system

section 2	**Quick Reference Guide** How to run the several WAQUA subsystems
section 3	**User's Guide WAQWND** Input description WAQWND
section 4	**User's Guide pre-processor WAQPRE** Input description WAQPRE
section 5	**User's Guide for Parallel WAQUA/TRIWAQ and for Domain Decomposition** Possibilities and limitations of domain decomposition; Input description COPPRE
section 6	**User's Guide processor WAQPRO** Input description WAQPRO User routines
section 7	**User's Guide SIMPAR** Input description SIMPAR
section 8	**User's Guide SLIB3D** Input description SLIB3D

Although the User's Guides describing the separate subsystems make some explicit references to section 1 (General Information), their dependencies on it are literally too numerous to mention. If a section in a specific User's Guide is not clear, the reader is advised to see the corresponding chapter in General Information.
The WAQUA system as described in this User's Guide WAQUA only deals with SIMONA based subsystems.

Chapter 2

About SIMONA

In this chapter SIMONA is described. In section 2.1 the various parts of a SIMONA program system are treated. In section 2.2 the SIMONA data storage is discussed. The input file for a SIMONA program is described in section 2.3.

2.1 Introduction

Each SIMONA program system consists of three parts: a pre-processing part, a processing part, and a post-processing part.

The three parts of a SIMONA program system have different functions:

pre-processing	In the pre-processing part all actions are taken that are needed before the actual computation can start, such as reading user input from the input file, defining the geometry, and filling parameters, variables and arrays with initial values. The pre-processor consists of an application independent part and an application dependent part. First the application independent part, which is provided by SIMONA, reads the complete input file, checks its syntax and stores the data in core. Next the application dependent part does the interpretation of the data, checks the validity of the input, derives other initial data and writes the input to an SDS file (usually in overwrite mode (see section 2.2)).
processing	In the processing part the actual computation takes place. All data needed are read from the SDS file. Results of the computation are written to the SDS file (usually in append mode (see section 2.2)).
post-processing	In the post-processing part the output is taken care of. The data are read from the SDS file and presented in a well-ordered way in prints and/or graphics. At this moment this part is not available.

2.2 SIMONA data storage

The three parts of a SIMONA program communicate with each other through a SIMONA data storage file (SDS file).

There are three types of access to an SDS file: read access, overwrite access, and append access. These types are used to protect the integrity of the data on the SDS file.

If a program is writing in overwrite mode to an SDS file, no other program can obtain access to that SDS file. If a program is writing in append mode to an SDS file, other programs can obtain read access to that SDS file, but no program can obtain write access to that SDS file (neither in overwrite mode nor in append mode). If a program is reading from an SDS file, other programs can obtain read access to that SDS file and one program can obtain append access to that SDS file, but no program can obtain overwrite access to that SDS file. The administration whether programs are reading/writing to an SDS file is stored on the SDS file itself. This internal administration will be incorrect when a program that had some kind of access to that SDS file has crashed. This might disable other programs to get the type of access to that SDS file they need. This problem can be corrected with the SIMONA utility SIRECOVR.

2.3 The input file

The SIMONA application independent preprocessor imposes the following syntax on the input file.

The input file is a structured ASCII file. From the input file only the first 258 columns are read. The structure of the input file is designed by the application developer. He has divided the input into separate blocks, which themselves may be subdivided into sub-blocks etc. in a hierarchical order. The distinction between the (sub-)blocks is done through the use of keywords. The names of the keywords as well as their hierarchical relation are chosen by the application developer.

Block

A block is defined as all information (including sub-blocks) between two keywords of the same level. The sequence of (sub-)blocks related to keywords of the same level can be chosen arbitrarily. Each (sub-)block can be specified only once, and all data concerning that (sub-)block must be given at that place. It is not allowed to add data for the same (sub-)block further down in the input file. A keyword defines a block. Therefore a defined block has to contain information, i.e. may not be empty. This principle means that at least one sub-keyword of a defined block ought to be defined in the input file, even when all these sub-keywords are optional.

Five types of items can be distinguished:

Keywords

1) Keywords:

A keyword is defined by a sequence of letters and underscores only. Keywords are case insensitive. The number of significant characters is imposed by the application developer, thus enabling the user to abbreviate lengthy keywords or extend unclear keywords.

reserved keyword

There are some reserved keywords which are used to forward special information. The following reserved keywords are available:

include

- INCLUDE:
Utilizing the keyword INCLUDE the user specifies that a data file has to be included in the input file. For the included file the same rules apply as for the original input file, except for the fact that it may not contain any includes itself.
Usage (with text in [square brackets] being optional):
include [file =] *filename*
The include file name can contain an explicit path name. The use of any indication of a parent directory ('..') is allowed.

insert

- INSERT:
Utilizing the keyword INSERT the user specifies that a file has to be inserted in the input file. In the inserted file however, only data are allowed, eventually with a number of heading comment lines. The user specifies the contents of the inserted file by:

nheadings	= –: number of heading lines, default = 0
number	= –: number of values to be read (mandatory)
format	= –: format to be used to read the values.

The following values are available for format:

- 0: free format (default)
- 1: unformatted
- 2: (10i6)
- 3: (10x, 12F5)
- 4: (16F5.0)
- 5: (12F6.0)
- 6: (10F8.0)

Note: the use of FORTRAN format implicates that tabulators are not allowed as separators in the insert file.
Usage:
insert [file =] *filename*, [nheadings = ..], number = .., [format = ..]
The insert file name can contain an explicit path name. The use of any indication of a parent directory ('..') is allowed.

SET keyword

- SET:
With the keyword SET the user may set one of the following control flags or variables:

set echo:	echo input from present position
set noecho:	suppress echoing from present position
set nowritesds:	suppress writing results of pre-processing to the SDS file (supported by SIMONA applications unless stated otherwise)
set maxwarn:	<number>, defines the maximum number of warnings that will be printed, default: the setting in the SIMONA environment file
set maxerror:	<number>, defines the maximum number of errors that will be printed, default: the setting in the SIMONA environment file

Numerical data 2) Numerical data:
A number is defined in the FORTRAN way, i.e. a row of symbols from the group . 0 1 2 3 4 5 6 7 8 9 + -, i.e. dot, digits and signs, ordered in the usual way: + - are only allowed at the first position and only one of both, furthermore, only one decimal point is allowed in the total row while all digits may be repeated unrestrictedly; the row may be extended with d, e, D or E as in the standard FORTRAN format.

Character strings 3) Character strings:
A character string is everything that is placed between a begin- and end-quote: ' and '. A quote is not allowed in the string itself.

Comment lines 4) Comment lines:
The user may add comment lines in the input file. These comment lines are ignored by the program. Such comments may be of great importance because they can clarify the meaning of the keywords used. All text after a hash character (#) in an input line is considered to be comment (as is all text beyond the 258th column).

Separation characters

5) Separation characters:
The following characters are recognized as separation marks, i.e. they do not have a meaning themselves but do separate other items:

space	
back-slash	\
slash	/
equal sign	=
open parenthesis	(
close parenthesis	)
comma	,
colon	:
semi-colon	;
tabulator	
end of input line (i.e. position 259)	

The user is advised to make full usage of the possibilities to structure his input (by using separation characters, comment lines, and indentation). This will result in an input file that is clear and readable.

Chapter 3

About WAQUA

The text of this section about WAQUA originates for a large part from Rand Corporation Working Draft WD-822-Neth, written by C.N. Johnson, A.B. Nelson and M.C. Fujisaki and later edited by J. Vincent and G.J. Bosselaar for earlier versions of the WAQUA user's guide.

3.1 Introduction to the system

This system is called WAQUA. It is used for two-dimensional hydro-dynamic and water quality simulation of well-mixed estuaries, coastal seas and rivers. As documented here, the system consists of a pre-processing program, a program for actual simulation and a number of post-processing programs for graphics output and prints. The major part of the system is written in FORTRAN77, with a minimum of computer dependent C code. It has been used on several mainframe and midrange computers and workstations. It is large, complex, and expensive in computer resources.

3.1.1 Function of the system

simulation
: The system can simulate hydrodynamics in geographical areas which are not rectangular, and bounded by any combination of closed boundaries (land) and open boundaries. Open boundaries can drive the model by water levels, velocities, Riemann invariants, discharges or distributed discharges, given either as time-varying data or Fourier/harmonic functions of phase and amplitude at given fre-quencies, or as a table, relating discharge with water level values. The system accounts for sources of discharge, such as rivers or outfalls, for tidal flats, for islands and dams, movable barriers or sluices and weirs. The water-quality computation handles several constituents.

graphics
: Such a simulation can generate great numbers of numbers. The second function of the system is to display computations in a highly visible way. Graphics include maps and time histories. Maps show computations for a point in time over the entire water body. Time history graphs show computations across time at one point in space. Time histories can also show computed and observed data together, for verification of a model, or to show the effect of proposed or hypothetical changes in the environment.

data management
: Some other functions of WAQUA can be grouped under the heading of data management. Input data are checked by a program and printed in a (future) input report for easy inspection by the user before the actual simulation, to minimize costly errors. The input report will, together with the input file, be an effective documentation of the model.

flexibility
: Also, the system is flexible so that small models can be run on small computers although large models require large computers (especially for the simulation itself). The program sizes vary with the size of the model, and the model can disregard various features of the system, including water-quality computation. Lastly, the system simplifies the use of the computer so that the user can concentrate on the research problem.

3.1.2 Features

simulation steps	When the system is fully used, the following steps are involved. The user sets up the data for a model by the use of the documentation and input file of another model. Then the user has that data checked, documented by print output, and filed by program WAQPRE. Then the simulation itself, program WAQPRO, is run, generating print of computations and file output for postprocessing.
graphics	Displays or high-quality plots are an indispensable part of the system to show a large volume of information visually. Since the simulation deals with two dimensions of space plus the time dimension, there are two kinds of displays, maps and time histories. Maps are generally snapshots, or pictures throughout space at a point in time. Time histories are time exposures, generally for a point in space or a cross-section. These types of postprocessing are possible with the WAQUA postprocessing programs at the moment available.
reports	Two features of reports are mentioned here: self-documenting reports and separated run messages. The self-documenting reports are those that print input data and control parameters, especially the (future) preprocessor (WAQPRE) input report. These minimize the user's dependence on separate, written documentation.
messages	Run, error, and warning messages from programs can be printed separately from reports, so that the success of a run can be determined without searching through a large volume of print. This is especially useful where runs are made from remote terminals and printing or viewing facilities are slow or otherwise inconvenient.
modularity	The water-quality feature of the system can be bypassed by giving no constituents. Likewise a model can omit time-varying open boundaries, Fourier-driven openings, outfalls or constituent sources, in fact any feature except the basic hydrodynamics.

model types

The user of the WAQUA system may use four types of grid systems:

- rectilinear coordinates;
- curvilinear coordinates;
- spherical coordinates;
- generalized spherical coordinates.

The choice between the above mentioned ways of modelling an area should be made in a very early stage of the modelling project because it is vital for all model input preparing activities.
The choice is governed by the characteristics of the area to be modelled.
A curvilinear grid however may be more favourable in cases where the area of interest is much smaller than the complete model area, or when the physical characteristics of the area demand small grid sizes on certain locations but allow at the same time larger grid sizes in other spots. This may be the case in river applications when an extensive winterbed is present.
Rectilinear/curvilinear models covering a relatively large part of the globe lose accuracy due to the sphere shape of the earth. For such models spherical coordinates is a good option.

space varying wind and pressure

In general WAQUA models cover a small area and spatial distribution of wind speed, wind direction and air pressure can be neglected. Spherical coordinate models however cover a considerable area where air pressure as well as wind speed and direction vary. In those cases WAQUA allows the user to use the space varying wind and pressure feature.

user routines

WAQUA offers the user possibilities to simulate dissolved substances. Constituents such as salinity, pollutants or algae can be modelled. As long as users deal with conservative dissolved substances, standard input facilities of the WAQUA system are sufficient.
Relations/reactions between these substances however should be modelled by means of a user defined 'user routine'. The system provides the heading of this routine. The user is responsible for programming his own reaction kinetics.

3.1.3 Modelling with WAQUA

the art of modelling

First of all, the system is not fool-proof; its answers are not necessarily realistic because its inputs require considerable judgement. Modelling is an art to be learned, beyond the knowledge of physical characteristics of water bodies. The limitations of digital computers and numeric simulation need to be understood. Potentially, after many computations, truncation errors can accumulate to a serious inaccuracy, due to a limited number of significant digits computers retain.

Fortunately, a model has been run for about 60 days with this system and there was no serious cumulative error. Many of the inputs, such as diffusion coefficients, must be determined by judgement and experimentation. A model should be verified, that is, matched with observed data. Also, sensitivity tests should be made to obtain an impression of the effect that changing parameters has on computational results.

model dimensions

Secondly, the water body should be of a practical size, to be simulated for a practical length of time. A model can get too large for available computers or prohibitive in cost. The basic size of a model is the number of grid spaces, and grid spaces should be small enough for some accuracy. Waves should not be short compared to grid space size, especially where eddies may be generated.

open boundaries

Long open boundaries should be placed at the edge of the grid. For a rectangular grid this implies m = 1 or m = MMAX or n = 1 or n = NMAX, but for a curvilinear grid m = 1 and n = 1 are not allowed (see also paragraph 3.2.1.3, computational grid tuning and open boundaries). The advective terms normal to the boundary are set to zero for a more stable boundary computation.

well mixed basins

An estuary, coastal sea or river can be modelled appropriately if it is well mixed that is, if one can assume no vertical velocity and no horizontal velocity gradient from surface to depth. In that case, no depth layering is required, and computation can be made on a horizontal grid. The grid distance is in the range of 20 to 10,000 meters, and the depth is on the order of 10 meters. The grid dimensions are on the order of 100 by 100.

The basic computation is of water movement between grid points and water level at each grid point. Optionally, the simulation will compute transport of constituents, such as salinity, carried with water movement and concentration of constituents at each grid point. Also with constituents computation no vertical gradients can be simulated.

computational grid	Within this grid, an area can be designated in which the computations are made. This area is called the computational grid. Computations at all points on the computational grid are performed every time step (where a time step is typically 1 to 2 minutes) for a simulated time of several days, usually.
monitoring	Computations throughout the field (the computational grid) may be printed and/or saved for later mapping (on the SDS file) at by the user specified intervals (typically every 1 to 12 hours of simulated time). To deal with the volume of computation and show connected histories of change (in water level, for example), a relatively few grid points (10 to 50) are chosen as stations. Computed values at stations are printed and saved more frequently (typically every 5 to 10 minutes) on the SDS file for later display as time histories.
limitations	The water body should be well-mixed (with vertically integrated velocity and concentrations) for two-dimensional simulation. This is only one example of the physical and numerical limitations, to indicate that a thorough knowledge of the system is required for its successful use. Some detailed limitations are listed in program user's guides, especially the WAQPRE and WAQPRO guides. There are no upper limits on the numbers of grid points, open boundaries, sources, dams, barriers, islands, stations, or cross-sections, or in the complexity of the shape of the computational grid. However, the number of grids points in both the x- and y-direction should be at least three (thus NMAX $\geq$ 3 and MMAX $\geq$ 3).
running programs	The information dealing with running the programs is available in the Quick Reference Guide to WAQUA.
computational methods	Some familiarity with the computational procedures and techniques used in WAQPRE and WAQPRO will help in designing a model. While this subject will not interest all users, some knowledge of the sequence of operations may be valuable if difficulties arise during a simulation. In most sections of chapter 3.4: General hydrodynamics, that describe the different properties of the system, subsections with information on computational methods are included.

3.2 Geometry

3.2.1 Computational grid

3.2.1.1 Computational grid structure

WAQUA grid

The WAQUA grid is illustrated in fig. 3.1. In modelling an estuary coastal sea or river, a rectangular geographical area is chosen, where its edges need not correspond with north, east, south and west. A grid is laid on the rectangular area, where the square grid space size in meters is chosen, and the number of grid spaces in the two dimensions, NMAX and MMAX, are chosen. The grid direction from left to right corresponds to the m-direction for grid space subscripts and to the u-velocity. The upward direction corresponds to the n-direction for subscripts and to the v-velocity.

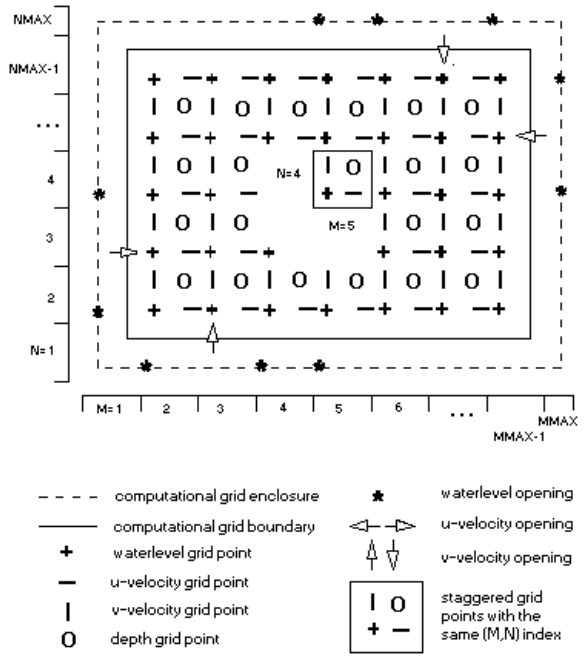


Figure 3.1: Default computational grid with arbitrary openings

Note: There are effectively one fewer row and column of velocities then water levels on the right and the top.

type of grid

Either a rectangular grid, a spherical grid or a curvilinear grid may be chosen. In case of a curvilinear grid a file with curvilinear coordinates has to be provided (see section 2.5.1.2 in the user's guide WAQPRE).

staggered grid

Four basic physical properties pertain to each grid space: water level, depth, u-component of velocity and v-component of velocity. In the computation there are four grids, one for each of these properties, identical in size and shape but staggered by half a grid space in one direction or both directions. The lower left corner of the chosen geographical area coincides with water level grid point (1, 1). Depth grid point (1, 1) is translated half a grid space to the right and above, so that depth grid point (1, 1) falls midway between water level grid points (1, 1) and (2, 2). The u-velocity grid point (1, 1) lies half a grid space to the right, so that it lies midway between water level points (1, 1) and (m = 2, n = 1). The v-velocity grid point lies midway between water level points (1, 1) and (m = 1, n = 2).

3.2.1.2 Basic grid principle

The staggered grid, which forms the basis of the WAQUA system, implies that a modelling system organised in this way can be regarded as consisting of a large number of linked, column shaped, volumes of water.

The corners of these volumes are the depth points in the grid. For rectilinear grids this implies that the base of the column is a square. For other WAQUA grids (spherical and curvilinear) the base of the columns is a quadrangle.

Each volume of water has four sides through which water may flow in or out of the volume. Referring to the principle of:

in = out + storage

it can be easily understood that adding the four flows through the four sides results in a rise or fall of the water level inside the volume.

Fig. 3.5 gives a good impression of the implications of the staggered grid principle.

computation of depth values

Usually the average depth is used on the boundaries of a grid cell for the computation of the flow in or out of the control volume. The water levels are averaged perpendicular to the cell boundary and the depths are averaged along the boundary; see fig. 3.2.

This approximation of the cross section normally ensures an accurate computation of the mass transport through the boundary of the control volume. However, in close vicinity to steep bottom gradients, as at drying flats and the change-over of winter-bed to summer-bed, the average approach may lead to an inaccurate representation.

tiled depth approach

For those situations the so-called 'upwind'-approach is more suitable. This is discussed more in detail in the chapter on 'Drying and Flooding' (section 3.6).
For determining the depths we propose a so-called 'tiled depth'-approach. Then the depth at a velocity point is equal to the minimum depth at the water level points of the two surrounding control volumes.
Alternatively, it is also possible to specify the depths on input at water level points.

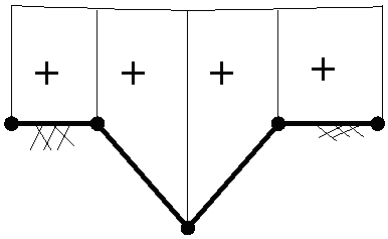


Figure 3.2: Wet cross section based for the "average approach" (depth in velocity point is the average of the depths in neighbouring depth points)

When using the tiled depth approach, it is possible to represent a small channel (narrow gully) of one grid cell width, see Fig. 3.2 and Fig. 3.3. Furthermore, the 'average approach' will lead in shallow areas to so-called 'screens' (in Dutch 'schotjes') in the tangential (or y-)direction, see Fig. 3.4. This is caused by the fact that bottom gradients in y-direction affect the drying behaviour in x-direction.

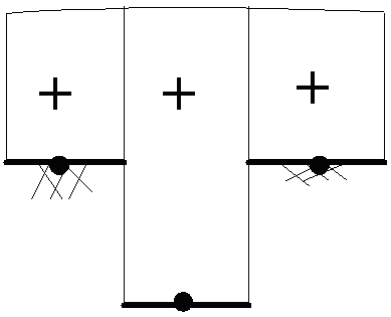


Figure 3.3: Wet section based on a "tiled depth approach" (depth in velocity point is the minimum of the depths in neighbouring depth points)

For a more detailed description we refer to the chapter concerning 'Drying and Flooding' (section 3.6).

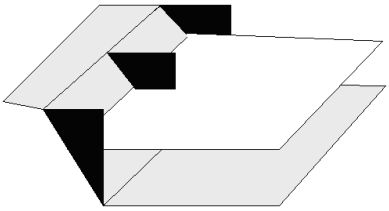


Figure 3.4: Screens in tangential direction occur when the depth in a velocity point is computed as the average depth value of the neighbouring depth points

3.2.1.3 Computational grid tuning and open boundaries

non rectangular comp. grid

The simulation program requires that open boundaries (e.g. tide openings) where measured tide or river data are given to drive the model, are on the edge of the computational grid. Where existing tide gauges are located so as not to fit a rectangle, a non-rectangular computational grid is fitted to the tide openings. Also, a computational grid may be defined, for the purpose of limiting computations, to an area of the grid that closely corresponds to the actual shape of the water body, for efficiency.

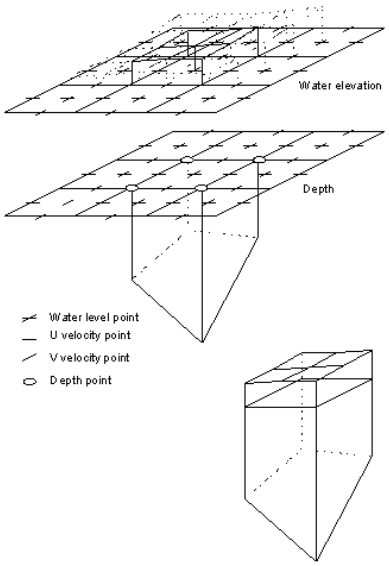


Figure 3.5: Control volume of the continuity equation of the finite difference representation in WAQUA

default computational grid

Previously tide level openings were always located at columns m = 1 and m = MMAX or rows n = 1 or n = NMAX, and the computational grid extended from m = 2 through MMAX - 1 and n = 2 through NMAX - 1 (rectangular). This is still the default, if no special computational grid is defined. Note that the 'default computational grid enclosure' is superimposed on all tide openings.

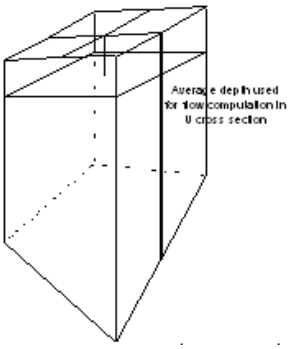


Figure 3.6: Flow cross section

computational grid enclosures

The computational grid within the rectangular grid is defined by one or more enclosures, superimposed on all water level openings. Each enclosure is a closed figure or polygon passing through (m, n) water level points, and falling just outside the computational grid, as the water level openings do. The computational grid, the enclosures and the type of boundaries (viz. water level, velocity, discharge or Riemann boundaries) are defined in more detail in the WAQPRE user's guide.

water level openings

With the extension of the simulation program to handle cases like the North Sea, the same idea becomes more complex. A rectangular grid is still chosen to cover the area to simulate. Tide level openings are still related to (m, n) points on the grid, but as columns, rows, or now as diagonals that are a multiple of 45 degrees. In other words, where the two ends of an opening are at (m1, n1) and (m2, n2) for diagonals, |m2 - m1| = |n2 - n1|.

Note: In case of a curvilinear computation tide level openings situated on the m = 1 column or the n = 1 row are not allowed.

velocity openings

Where velocities rather than water levels have been measured, velocity openings may be defined. Velocity openings are single points or rows or columns on the (m, n) grid, but not diagonals, since velocity openings are associated with horizontal or vertical components of velocity. Velocity openings also lie just outside the computational grid, but within the defined grid enclosure. In the space-staggered grid, velocity points lie between water level points (see fig. 3.1). For a given (m, n) index its velocity points '-' and '|' lie to the right and above the '+' water level point. This means that for velocity openings on the left or bottom of the computational grid, their (m, n) indices are the same as those included in the enclosure. However, on the right, a velocity opening is assigned one m-column to the left of the enclosure, and velocity openings at the top of the computational grid are assigned one n-row below the enclosure.

discharge openings

If the discharge rates at some points of the computational grid enclosure are known, discharge openings may be defined. Local velocities are derived from discharges according to the formula:

$$v = \frac{Q}{H \cdot \Delta x}$$

where:

v	=	flow velocity (m/s)
Q	=	discharge rate (m^3/s)
H	=	water depth (m)
Δx	=	width of the opening (m)

A special form of discharge opening is an opening with a so-called 'automatic' distributed discharge: a user specified total discharge through an opening is distributed over the points along the opening, accounting for local water depth and bottom friction.

Restrictions for the discharge openings are the same as for the velocity openings.

Riemann openings

When the combination of both water levels and velocities is known, a Riemann invariant boundary condition can be assigned to an opening. A further description can be found in section 3.4.2.3.

Restrictions for the Riemann openings are the same as for the velocity openings.

Q-H openings

When the relation between water level and total discharge through an opening is known (e.g. in rivers), the water level at the opening can be computed during the computation from the total discharge. Q-H type of openings are specified with boundary points located as for water level openings, with the restriction to only horizontal or vertical open boundaries (like velocity openings).

3.2.2 Physical geometry

dry points or dams	The modeller may define certain grid points to be permanently dry points, overriding the description of depth throughout the field. This provides a means to make a dam or causeway through the water body with considerable depth at either side.
screens	The modeller may define screens at certain velocity points. This enables the modeller to create dams without width, to simulate for instance groynes, which have a width small compared with the grid size.
weirs	The modeller may define certain grid points to be weirs, which are fixed, non-movable constructions causing energy losses due to constriction of flow.
barriers or sluices	The modeller may define certain grid points to be sluices or barriers, which are a time-varying constriction on velocity in the u-direction or the v-direction.

3.3 Model types

3.3.1 Rectilinear models

3.3.1.1 General

Originally the WAQUA system could only cope with rectilinear grids. In large parts of the user's guide a rectilinear grid is assumed, but in most cases the documentation is also valid for curvilinear and spherical grids.

For a rectilinear grid there is a simple relation between (m, n) coordinates and (x, y) locations. The input and output of the subsystems however refer in most cases to the (m, n) coordinates.

Defining the rectilinear grid can be carried out easily. Numerically the rectilinear model equations are simpler and because of that the model is faster during computation. The mathematical accuracy can be assessed easier than for the other grid types.

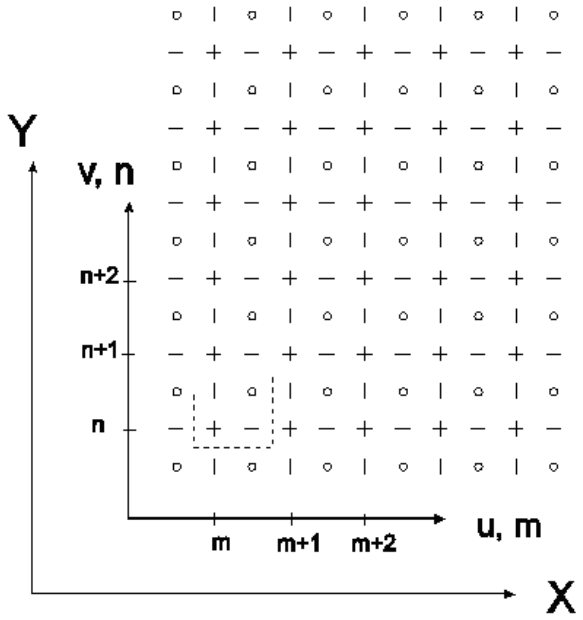


Figure 3.7: Rectilineair grid

The figure gives an impression of the regular structure of a rectilinear model. There is a straight forward relation between the (m, n) model coordinates and an (x, y) reference.

3.3.1.2 Mathematical description differential equations

The following equations govern the rectilinear WAQUA computation:

- shallow water equations

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - f\,v + g\frac{\partial \zeta}{\partial x} + gu\frac{\sqrt{u^2+v^2}}{C^2(h+\zeta)} = \frac{\rho_a\, C_d\, W_x \sqrt{W_x^2+W_y^2}}{\rho_w(h+\zeta)} + \quad \nu \quad \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + f\;u + g\frac{\partial \zeta}{\partial y} + gv\frac{\sqrt{u^2+v^2}}{C^2(h+\zeta)} = \frac{\rho_a\, C_d\, W_y \sqrt{W_x^2+W_y^2}}{\rho_w(h+\zeta)} + \quad \nu \quad \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)$$

$$\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x}(Hu) + \frac{\partial}{\partial y}(Hv) = 0$$

where:

u,v	=	components of depth mean current $\tilde{u}$
ζ	=	water elevation above plane of reference
h	=	water depth below plane of reference
H	=	h + ζ (refer to Fig. 3.8)
f	=	parameter of Coriolis
g	=	acceleration due to gravity
C	=	coefficient of Chezy to model bottom roughness
W_x, W_y	=	components of surface wind velocity $\tilde{W}$
C_d	=	wind drag coefficient
ρ_a, ρ_w	=	density of air and water
υ	=	eddy-viscosity coefficient

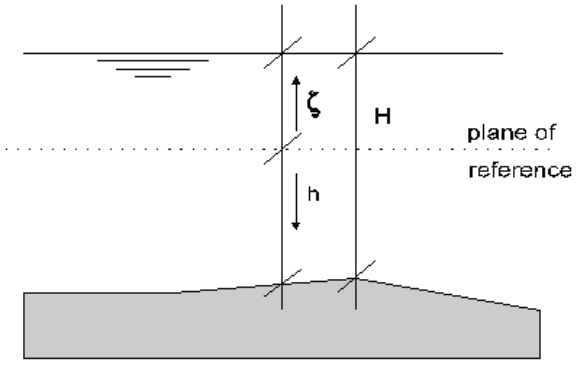


Figure 3.8: Layer of water = water depth + water elevation

- dissolved constituents:

$$\frac{\partial}{\partial t}(Hc) + \frac{\partial}{\partial x}(Huc) + \frac{\partial}{\partial y}(Hvc) = \frac{\partial}{\partial x}\left(HD_x\frac{\partial c}{\partial x}\right) + \frac{\partial}{\partial y}\left(HD_y\frac{\partial c}{\partial y}\right) + S_c$$

where:

C	=	constituent concentration
S_c	=	sources or sinks
D_x, D_y	=	diffusion coefficients in x- and y-direction

3.3.2 Curvilinear models

3.3.2.1 General

In the 1980's a curvilinear version of the WAQUA system was developed. In the curvilinear case computations are carried out on a non-equidistant grid, refer to Fig. 3.9.

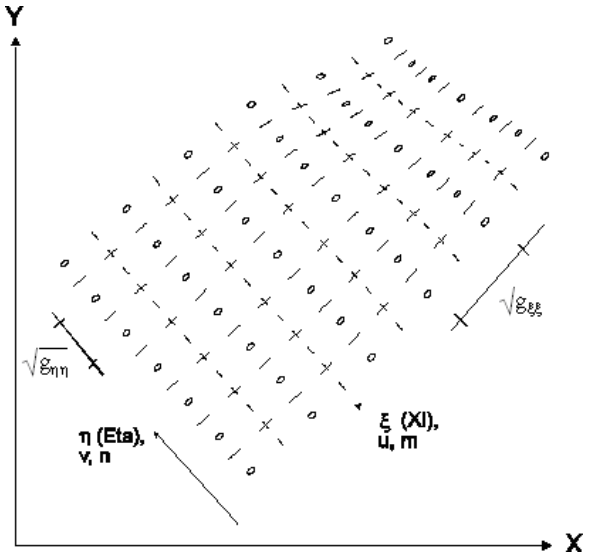


Figure 3.9: Curvilineair grid

The grid is assumed orthogonal. The degree in which the angles of the grid used do not match the ideal right angle (90 degrees) is an indication of the numerical error in the computation.

$\sqrt{g_{\eta\eta}}$ is defined in the u-velocity point.

$\sqrt{g_{\xi\xi}}$ is defined in the v-velocity point.

$$g_{\xi\xi} = \left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2$$

$$g_{\eta\eta} = \left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2$$

are the transformation coefficients from the cartesian system (x,y) to the orthogonal curvilinear system (ξ,η).

Rules of thumb to keep the numerical accuracy acceptable:

- grid size variations should be limited to a factor 1.3 for neighbouring grid cells;
- angles within grid cells should meet the condition:

100 * cos (angle) < 2 (deviation of approximately 1.2 degrees).

Cells not meeting this criterion should be examined by a model engineer.

Curvilinear grids may be applied in cases where the area of interest is much smaller then the complete model. Another case in which curvilinear models prove to be a good alternative occurs when the physical characteristics of the area involved show a great variety. An example of this phenomena is a Dutch river with a relatively narrow summer bed and a wide winter-bed. To model the flow (discharge) through the summer bed properly, a small grid size is required. The winter-bed however is shallow and small grid sizes are not necessary when modelling the discharge and the flow pattern in that area.

The curvilinear grid must be given to the WAQUA system as an input file, the so called RGF file. The format of this file is straight forward ASCII. The RGF file can be used as direct input for the WAQPRE subsystem.

The RGF file is the output of a grid generating process for which a grid generator program should be used. The WAQUA system is not equipped with a grid generator.

3.3.2.2 Mathematical description differential equations

The following equations govern the curvilinear WAQUA computation:

- shallow water equations:

$$\frac{\partial u}{\partial t} + \frac{u}{\sqrt{g_{\xi\xi}}}\frac{\partial u}{\partial \xi} + \frac{v}{\sqrt{g_{\eta\eta}}}\frac{\partial u}{\partial \eta} + \frac{uv}{\sqrt{g_*}}\frac{\partial \sqrt{g_{\xi\xi}}}{\partial \eta} - \frac{v^2}{\sqrt{g_*}}\frac{\partial \sqrt{g_{\eta\eta}}}{\partial \xi} - fv + \frac{g}{\sqrt{g_{\xi\xi}}}\frac{\partial \zeta}{\partial \xi} + gu\frac{\sqrt{u^2+v^2}}{C^2(h+\zeta)} = \frac{\rho_a C_d W_\xi \sqrt{W_\xi^2+W_\eta^2}}{\rho_w(h+\zeta)} + \frac{v}{\sqrt{g_{\xi\xi}}}\frac{\partial A}{\partial \xi} - \frac{v}{\sqrt{g_{\eta\eta}}}\frac{\partial B}{\partial \eta}$$

$$\frac{\partial v}{\partial t} + \frac{u}{\sqrt{g_{\xi\xi}}}\frac{\partial v}{\partial \xi} + \frac{v}{\sqrt{g_{\eta\eta}}}\frac{\partial v}{\partial \eta} + \frac{uv}{\sqrt{g_*}}\frac{\partial \sqrt{g_{\eta\eta}}}{\partial \xi} - \frac{u^2}{\sqrt{g_*}}\frac{\partial \sqrt{g_{\xi\xi}}}{\partial \eta} + fu + \frac{g}{\sqrt{g_{\eta\eta}}}\frac{\partial \xi}{\partial \eta} + gv\frac{\sqrt{u^2+v^2}}{C^2(h+\zeta)} = \frac{\rho_a C_d W_\eta \sqrt{W_\xi^2+W_\eta^2}}{\rho_w(h+\zeta)} + \frac{v}{\sqrt{g_{\eta\eta}}}\frac{\partial A}{\partial \eta} + \frac{v}{\sqrt{g_{\xi\xi}}}\frac{\partial B}{\partial \xi}$$

$$\frac{\partial \zeta}{\partial t} + \frac{1}{\sqrt{g_*}}\frac{\partial}{\partial_\xi}\left(Hy\sqrt{g_{\eta\eta}}\right) + \frac{1}{\sqrt{g_*}}\frac{\partial}{\partial_\eta}\left(Hy\sqrt{g_{\xi\xi}}\right) = 0$$

where:

$$A = \frac{1}{\sqrt{g_*}}\left[\frac{\partial}{\partial \xi}\left(u\sqrt{g_{\eta\eta}}\right) + \frac{\partial}{\partial \eta}\left(v\sqrt{g_{\xi\xi}}\right)\right]$$
$$B = \frac{1}{\sqrt{g_*}}\left[\frac{\partial}{\partial \xi}\left(u\sqrt{g_{\eta\eta}}\right) - \frac{\partial}{\partial \eta}\left(v\sqrt{g_{\xi\xi}}\right)\right]$$

and:

u,v	=	components of depth mean current $\tilde{\mathrm{u}}$
ζ	=	water elevation above plane of reference
h	=	water depth below plane of reference
H	=	$h + \zeta$
f	=	parameter of Coriolis
g	=	acceleration due to gravity
C	=	coefficient of Chezy to model bottom roughness
W_ξ, W_η	=	components of surface wind velocity $\tilde{\mathrm{W}}$
C_d	=	wind drag coefficient
ρ_a, ρ_w	=	density of air and water
υ	=	eddy-viscosity coefficient
$g_{\xi\xi}$, $g_{\eta\eta}$	=	transformation coefficients (see par. 3.3.1.1)
g_*	=	$g_{\xi\xi} \cdot g_{\eta\eta}$

- dissolved constituents:

$$\frac{\partial}{\partial t}(Hc) + \frac{1}{\sqrt{g_*}}\frac{\partial}{\partial \xi}\left(Huc\sqrt{g_{\eta\eta}}\right) + \frac{1}{\sqrt{g_*}}\frac{\partial}{\partial \eta}\left(Hvc\sqrt{g_{\eta\eta}}\right) =$$

$$\frac{1}{\sqrt{g_*}}\frac{\partial}{\partial \xi}\left(HD_\xi \frac{\sqrt{g_{\eta\eta}}}{\sqrt{g_{\xi\xi}}}\frac{\partial c}{\partial \xi}\right) + \frac{1}{\sqrt{g_*}}\frac{\partial}{\partial \eta}\left(HD_\eta \frac{\sqrt{g_{\xi\xi}}}{\sqrt{g_{\eta\eta}}}\frac{\partial c}{\partial \eta}\right) + S_c$$

where:

c	=	constituent concentration
S_c	=	sources or sinks
D_ξ, D_η	=	diffusion coefficients in ξ- and η-direction

3.3.3 Spherical coordinates models

3.3.3.1 General

When the area to be modelled covers a considerable part of the globe, the assumption of a flat (rectilinear or curvilinear) model is no longer valid. For such a model area a spherical coordinates WAQUA model can be defined, refer to Fig. 3.10.

The grid size is no longer defined in meters but in degrees. Users are advised to choose for spherical grids with an n-axes pointing North and an m-axes pointing East. Spherical grids with m- and n-axes not coinciding with longitude and latitude are possible but not convenient in use.

3.3.3.2 Mathematical description differential equations

The following equations govern the spherical coordinates WAQUA computation:

- shallow water equations:

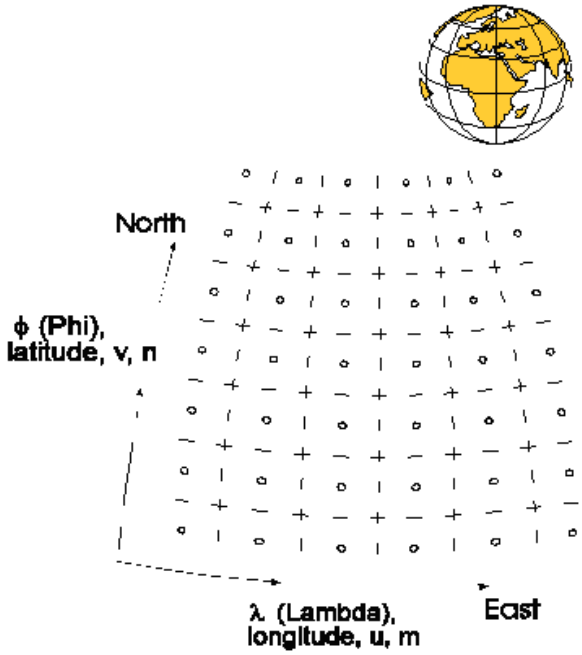


Figure 3.10: Spherical coordinates grid

$$\frac{\partial u}{\partial t}+\frac{u}{R\cos\phi}\frac{\partial u}{\partial \lambda}+\frac{v}{R}\frac{\partial u}{\partial \phi}-\frac{uvtg\phi}{R}-2\Omega v\sin\phi-\frac{g}{R\cos\phi}\frac{\partial \zeta}{\partial \lambda}+gu\frac{\sqrt{u^2+v^2}}{C^2(h+\zeta)}=\frac{\rho_a C_d W_\lambda\sqrt{W_\lambda^2+W_\phi^2}}{\rho_w(h+\zeta)}-\frac{1}{\rho_w R\cos\phi}\frac{\partial p_a}{\partial \lambda}$$

$$\frac{\partial v}{\partial t}+\frac{u}{R\cos\phi}\frac{\partial v}{\partial \lambda}+\frac{v}{R}\frac{\partial v}{\partial \phi}+\frac{u^2tg\phi}{R}+2\Omega v\sin\phi+\frac{g}{R}\frac{\partial \zeta}{\partial \phi}+gv\frac{\sqrt{u^2+v^2}}{C^2(h+\zeta)}=\frac{\rho_a C_d W_\phi\sqrt{W_\lambda^2+W_\phi^2}}{\rho_w(h+\zeta)}-\frac{1}{\rho_w R}\frac{\partial p_a}{\partial \phi}$$

$$\frac{\partial \zeta}{\partial t}+\frac{1}{R\cos\phi}\frac{\partial}{\partial \lambda}(Hu)+\frac{1}{R\cos\phi}\frac{\partial}{\partial \phi}(Hv\cos\phi)=0$$

where:

u,v	=	components of depth mean current $\tilde{u}$
ζ	=	water elevation above plane of reference
h	=	water depth below plane of reference
H	=	$h + \zeta$
Ω	=	angular speed of the earth's rotation
g	=	acceleration due to gravity
C	=	coefficient of Chezy to model bottom roughness
R	=	radius of the earth
W_ξ, W_η	=	components of surface wind velocity $\tilde{W}$
C_d	=	wind drag coefficient
ρ_a, ρ_w	=	density of air and water
p_a	=	surface atmospheric pressure

- dissolved constituents:

$$\frac{\partial}{\partial t}(Hc)+\frac{1}{R\cos\phi}\frac{\partial}{\partial \lambda}(Huc)+\frac{1}{R\cos\phi}\frac{\partial}{\partial \phi}(Hvc\cos\phi)=\frac{1}{R\cos\phi}\frac{\partial}{\partial \lambda}\left(HD_\lambda\frac{1}{R\cos\phi}\frac{\partial c}{\partial \lambda}\right)+\frac{1}{R\cos\phi}\frac{\partial}{\partial \phi}\left(HD_\phi\frac{1}{R}\frac{\partial c}{\partial \phi}\cos\phi\right)+S_c$$

where:

c	=	constituent concentration
S_c	=	sources or sinks
D_λ, D_ϕ	=	diffusion coefficients in λ- and ϕ-direction

Note: There is a strong relationship between the equations for curvilinear coordinates and the equations for spherical coordinates. Define a fixed Cartesian frame through the origin of the earth, then the spherical coordinates of a point $\vec{\mathrm{r}}$ on the earth's surface are given by:

$\vec{\mathrm{r}}$ = (x, y, z)T = (λ, ϕ, R)T

with:

x	=	R $\cdot$ cos ϕ $\cdot$ cos λ,
y	=	R $\cdot$ cos ϕ $\cdot$ sin λ,
z	=	R $\cdot$ sin ϕ.

The transformation coefficients are:

$$\sqrt{g_{\lambda\lambda}} = \left\| \frac{\partial \vec{r}}{\partial \lambda} \right\| = \left\| \begin{pmatrix} -R\cos\phi\sin\lambda \\ -R\cos\phi\sin\lambda \\ 0 \end{pmatrix} \right\| = R\cos\phi,$$

$$\sqrt{g_{\phi\phi}} = \left\| \frac{\partial \vec{r}}{\partial \phi} \right\| = \left\| \begin{pmatrix} -R\sin\phi\cos\lambda \\ -R\sin\phi\cos\lambda \\ \mathrm{Rcos}\phi \end{pmatrix} \right\| = R$$

Replacing ξ and η by:
$\xi = \lambda, \eta = \phi \Rightarrow \sqrt{g_{\xi\xi}} = \sqrt{g_{\lambda\lambda}}\sqrt{g_{\eta\eta}} = \sqrt{g_{\phi\phi}}$
Substituting these formulas in the equations for a curvilinear grid, the equations in spherical coordinates without Coriolis-force and viscosity terms are derived.

3.3.4 Models using space varying wind and pressure

For all three earlier mentioned model types the Space Varying Wind and Pressure (SVWP) feature is available. It enables the user to drive his model with an extensive set of time dependent and spatial distributed wind boundary conditions. Fields of wind directions, wind speeds or stresses and air pressures can be processed by WAQUA, while interpolating in time.

A (λ, ϕ) grid can be transformed into an (x, y) grid. If the pressure parameters are required in (λ, ϕ) form, they should be offered to the WAQUA system in their own (λ, ϕ) grid. The wind grid should have at least the size of the water movement and quality model. An overlap is allowed and even recommended.

The WAQUA system is not able to extrapolate wind related data to parts of the model which are not covered by the wind grid. Flat (rectilinear or curvilinear) models may be provided with a flat or spherical wind grid. The spherical models can only cope with spherical wind grids.

The file describing the time and space distributed wind and pressure data may be of considerable size. In most cases it is produced and delivered by the dutch meteorological office (KNMI). The file is transformed into a SIMONA SDS file by the WAQUA subsystem WAQWND. The name of this special SDS file should be stated in the WAQPRE input under 'general space_var_wind'. The format of the files involved is described in the WAQUA system's documentation.

Instead of wind speeds it is also possible to provide WAQUA with wind stresses. These stresses must have been computed as:

$$S_x = \rho_a \, C_d \, W_x \sqrt{W_x^2 + W_y^2}$$

$$S_y = \rho_a \, C_d \, W_y \sqrt{W_x^2 + W_y^2}$$

S_x, S_y	=	components of surface wind stress
W_x, W_y	=	components of surface wind velocity $\tilde{\mathrm{W}}$
C_d	=	wind drag coefficient
ρ_a	=	density of air

3.4 General hydrodynamics

3.4.1 Features

time step	The full time step in minutes is chosen and given to WAQPRO through WAQPRE. At the first half time step, u-velocities and resultant water levels are calculated and also separate v-velocities (explicit). At the second half time step, v-velocities and resultant water levels are calculated together with separate u-velocities (explicit). This means that u- and v-components of velocity are never completely synchronized in time (staggered time step). Also, the computation has alternating direction, that is, u-velocities are calculated with the m subscript varying faster, and the n subscript varies faster for v-velocity.
interpolation	All time-varying inputs are interpolated across time, and inputs at open boundaries are also interpolated across space.
time interpolation	Interpolation across time occurs for the time series at tide openings and at discharge points, which can be equidistant or not. The value at the beginning of interpolation is either the previous value given in the input, or the initial value (given or assumed) (see paragraph 3.4.2.3).
space interpolation	Interpolation across space occurs at open boundaries, where inputs are given at the two ends of the opening, and values at intervening grid points on the opening are calculated by interpolation. Space interpolation is done to tide level or velocity at Fourier or time-varying openings, and to constituent concentration and - only initially - constituent return time (see section 3.8).
tide openings	The main hydrodynamic forcing function is observed tide at open boundaries or tide openings. Historically the simulation has accepted water level values at a regular time interval. Extensions to the simulation accept not only velocity, Riemann or discharge values at a regular time interval but also Fourier functions of time-varying values of either water level, velocity, Riemann or discharge. For application of WAQUA to river hydraulics, boundary conditions can be specified as a total (upstream) discharge or as a relation between total (downstream) discharge and water level. The Fourier functions have any number of angular frequency components with corresponding phase and amplitude values. The simulation interpolates phase and amplitude across the space of the open boundary before transforming back to the time domain and interpolating across time.

wind interpolation	Time interpolation of global wind takes place by interpolating the polar components wind angle and wind speed. Time interpolation of space varying wind takes place by interpolating the cartesian components wind speed in the X-direction and wind speed in the Y-direction.
non-hydrostatic computation	TRIWAQ has the possibility to take non-hydrostatic effects for free-surface flows into account.

3.4.2 Computational methods

In this section an overview is given of the time advancement procedure of state variables and accuracy considerations. Furthermore, forcing functions that drive the simulation are mentioned, as well as the computation of time integrals. The drying and flooding procedures will be described in section 3.6.2. Chezy values, which are defined at velocity locations, are related to the Manning coefficient. A full account of the numerical method that is used in WAQUA is given in:

G. S. Stelling, On the Construction of Computational Methods for Shallow Water Flow, Rijkswaterstaat Communications no 35/1984.

3.4.2.1 Time advancement of state variables

state variables	The integration proceeds in increments of half time steps (half of TSTEP). All state variables are computed in each half time step on a staggered grid (see Fig. 3.1). The state variables are: the water level ζ, velocities u and v in x and y directions, respectively (initially zero), and constituent concentrations.
ADI scheme	The numerical method is a composite ADI scheme, where in the first half time step v is calculated separately from u and ζ. In the second half time step u is calculated separately from v and ζ. The method has as properties: • it is symmetrical over the x and y directions, • it attains its highest accuracy over every two half time steps, • it is mass conserving, • it is computationally efficient, • it is suitable not only for time-dependent problems, but also for steady state problems, • it is unconditionally stable.

accuracy

This last property however, does not imply that the time step can be chosen without bounds. The quality of the solution is also dependent on the boundary conditions, the forcing functions and other time-dependent input. Therefore, in practice the accuracy puts a limit on the time step (TSTEP). In many problems the Courant number C_f is a relevant measure. For equal spatial steps (dx = dy) it can be defined by

$$C_f = \Delta t \cdot \frac{\sqrt{2gH}}{dx}$$

where:

H	=	total water depth
C_f	=	Courant number
Δt	=	time step = TSTEP
g	=	acceleration of gravity
dx	=	spatial step (grid space)

Thus, an estimate for the maximum value of Δt can be determined. However, a maximum limit on Δt is problem dependent. In some cases the geometry can put a limiting factor on the time step Δt, because of accuracy reasons. There is no time smoothing present in the integration procedure.

3.4.2.2 Time integration

Time integrals, 2D simulation

During the simulation 'time integrals' may be computed over one or more successive periods. For each integration period five time-integrated flow-related integrals are computed, which results in the following variables:

$$\text{DEPINT} = \int_t^{t+T} (H + \zeta) \cdot dt$$

$$\text{UPINT} = \int_t^{t+T} u \cdot dt$$

$$\text{VPINT} = \int_t^{t+T} v \cdot dt$$

$$\text{DISUNT} = \int_t^{t+T} (H + \zeta) u \cdot dt$$

$$\text{DISVNT} = \int_t^{t+T} (H + \zeta) v \cdot dt$$

where:

DEPINT	=	time-integrated total depth at W(ater) L(evel) grid location
DISUNT	=	time-integrated discharge at u-grid location
DISVNT	=	time-integrated discharge at v-grid location
H	=	water depth below plane of reference, (time-invariant)
ζ	=	momentary water elevation
T	=	integration period
u	=	momentary u-velocity component
UPINT	=	time-integrated u-velocity at u-grid location
v	=	momentary v-velocity component
VPINT	=	time-integrated v-velocity at v-grid location

ADI staggered time integration

In a WAQUA-simulation the computation is done, using an ADI staggered time integration method over two half time steps, so not all primary data are available at the same time:

```
ζ,v       ζ,u        ζ,v        ζ,u
I---------I---------I---------I----- ....
t   t + 0.5 · dt  t + dt  t + 1.5 · dt
```

Therefore (with k a non-negative integer):

- DEPINT is computed using all ζ-values at t = t + 0.5 · k · dt; method: trapezoidal rule;
- UPINT is computed using all u-values at t = t + (k + 0.5) · dt; method: rectangle rule;
- VPINT is computed using all v-values at t = t + k · dt; method: trapezoidal rule;
- DISUNT is computed using ζ- and u-values at t = t + (k + 0.5) · dt; method: rectangle rule;
- DISVNT is computed using ζ- and v-values at t = t + k · dt. method: trapezoidal rule.

Output

The time integrals are written as time-variant array to the SDS file at the end of each integration period. The ending time of each integration period is added as time attribute for the integrals.

Apart from the time the momentary water levels and an administrative array are written. The administrative array contains time information for all time integrals: starting time of the integration period, ending time of the integration period and the integration period duration. The momentary water levels are included for later appliance of an 'integrated continuity equation'. Therefore the momentary water levels are also added by an extra write at t = TFINT (starting time integration process).

Due to the restart option (see WAQPRE user's guide), time integrals are also written according to the restart frequency. These integrals on the SDS file can only be used for restart purposes.

Restrictions/reminders

- The integrals are computed at all grid points, including dry and non-computational points. There are two exceptions to that: DISUNT is set to zero for line m = MMAX and DISVNT is set to zero for line n = NMAX. The reason for that is that not all necessary data are available at those grid locations.
- Permanent restart (see WAQPRE user's guide) is only allowed under the following restrictions:
- the integration interval may not be changed, so

  TIINTEGR$_{old}$ = TIINTEGR$_{new}$

  where:

  TIINTEGR$_{old}$ = time interval to write integrals for old simulation
  TIINTEGR$_{new}$ = time interval to write integrals for new simulation
- the integration starting time of the new simulation must be equal to an integration starting time of the old simulation, so:

  TFINTEGR$_{old}$ = TFINTEGR$_{new}$

  where:

  TFINTEGR$_{old}$ = first time to write integrals for old simulation
  TFINTEGR$_{new}$ = first time to write integrals for new simulation
  k = an integer > 0

Time integrals, 3D simulation

During a TRIWAQ simulation 'time integrals' may be computed in the same way as in the 2D case (actually, the same computational routine is used in both cases). This time each integration period seven time-averaged flow-related integrals are computed, which results in the following variables:

$$\text{ZKINT}_k = \int_t^{t+T} ZK_k \cdot dt$$

$$\text{UPINT}_k = \int_t^{t+T} U_k \cdot dt$$

$$\text{VPINT}_k = \int_t^{t+T} V_k \cdot dt$$

$$\text{WINT}_k = \int_t^{t+T} \omega_k \cdot dt$$

$$\text{WPHINT}_k = \int_t^{t+T} Wphys_k \cdot dt$$

$$\text{DISUNT}_k = \int_t^{t+T} U_k H_k \cdot dt$$

$$\text{DISVNT}_k = \int_t^{t+T} V_k H_k \cdot dt$$

where:

ZKINT_k	=	time-averaged position of layer interface with index k at W(ater)L(evel)-grid location
UPINT_k	=	time-averaged u-velocity with index k at u-grid location
VPINT_k	=	time-averaged v-velocity with index k at v-grid location
WINT_k	=	time-averaged Omega-velocity with index k at WL-grid location
WPHINT_k	=	time-averaged Wphys-velocity with index k at WL-grid location
DISUNT_k	=	time-averaged discharge at u-grid location with index k
DISVNT_k	=	time-averaged discharge at v-grid location with index k
ZK_k	=	momentary position of layer interface with index k
U_k	=	momentary U-velocity component of layer k
V_k	=	momentary V-velocity component of layer k
ω_k	=	momentary Omega-velocity component of layer k
$Wphys_k$	=	momentary component of physical vertical velocity of layer k
H_k	=	momentary thickness of layer k
T	=	integration period

ADI staggered time integration

Like in the WAQUA-simulation, the computation in a TRIWAQ-simulation is done using an ADI staggered time integration method:

ZK_k, V_k, ω_k ZK_k, U_k, ω_k ZK_k, V_k, ω_k ZK_k, U_k, ω_k

```
I-----------I-----------I-----------I---.....
t          t+0.5dT      t+dT      t+1.5dT
```

With i a (non-negative) integer:

- UPINT_k, VPINT_k, DISUNT_k and DISVNT_k are computed in the WAQUA simulation.
- ZKINT_k, WINT_k and WPHINT_k are computed using all values of ZK_k at all t = t + 0.5 · i · dt using a trapezoidal rule.

Lagrangian displacements

The general idea In this section a description is given of the option in WAQUA/TRIWAQ concerning the calculation of Lagrangian displacements of points. In the standard hydrodynamic calculation, u- and v-velocities are calculated at fixed positions at a certain time. So, if a water mass

is released at a grid point, the displacement in a time step can be calculated using the velocities that are given in that grid point. The problem is that the point will generally not be displaced to another grid point but will end up somewhere in the middle of the grid cell. At that position no information about velocities is available and it is not straight forward to calculate a displacement for the next time step. The above mentioned option enables us to calculate velocities in the grid cell itself with some interpolation technique. These velocities are calculated using the velocities of the adjacent grid points. In this way the displacement of a water mass can be calculated by means of the velocities at the actual position of the water mass. This explains the word Lagrangian: velocities are not calculated at fixed positions in space but at a position moving with a certain water mass.

A quantity that also has to be stored, is the relative position of the released point within the grid cell at the end of every time interval. For this position will be the starting point of the displacement over the next time interval.

Displacements are calculated with the same frequency as velocities, every half time step, which makes the method second order accurate in time.

The user can give the starting time T_{start} and the end time T_{end} of period over which displacements have to be calculated. The time interval T_{int} over which displacements are calculated has to satisfy:

$k_1 \times T_{int} = T_{end} - T_{start}$

k_1 some integer value

This time interval T_{int} also has to be a multiple of the hydrodynamic time step dt:

$k_2 \times dt = T_{int}$

k_2 some integer value

The user has two possibilities concerning initialization at the start of the time interval T_{int}:

- points can be initialized at the beginning of each interval T_{int} and will be replaced to their starting positions. In this way the user is enabled to calculate displacements during separate time intervals.
- point will start at the beginning of an interval T_{int} from the position they have reached the previous time interval. This gives the possibility to calculated trajectories over a longer period which are quite smooth.

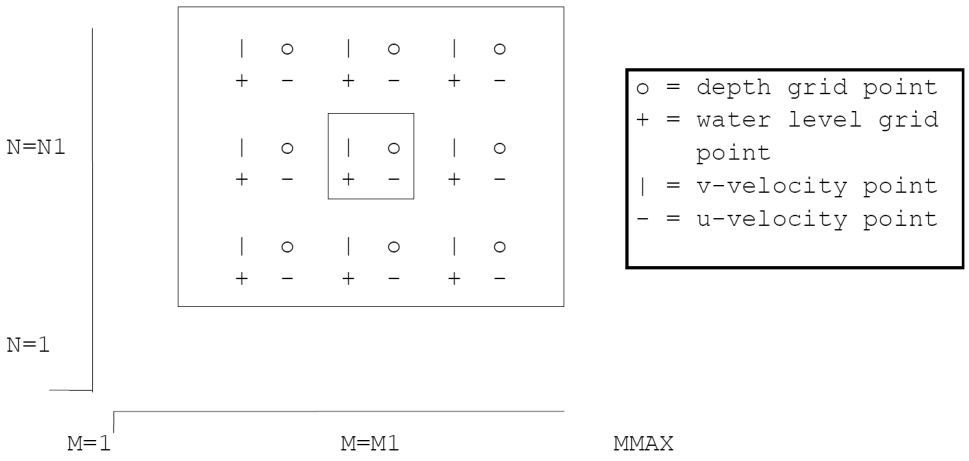
The calculation of displacements in a Lagrangian way can have great advantages when one is interested in transport phenomena in areas in which the flow pattern is strongly non-uniform. By following a water mass through the grid cells, the trajectories along which the masses are displaced are (numerically) better reproduced.

Starting position of the Lagrangian trajectories When this option is used, a point is released in every 3D-cell of the hydrodynamic grid, and an extra point is released at the bottom. So, when N

layers in the vertical are used, N+1 points are released per grid column.

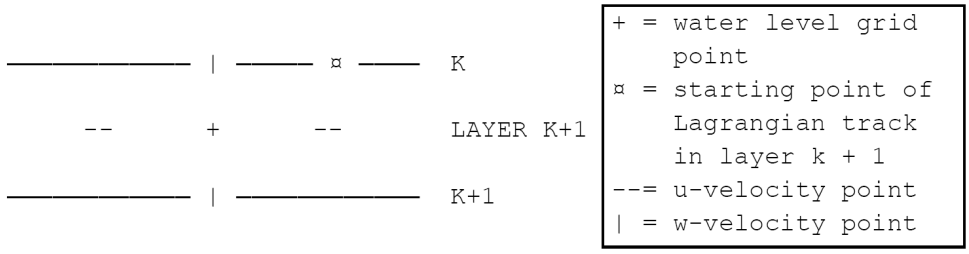
The starting position within the cells is in the depth points:

Starting position in the horizontal grid:



A point with index (M,N) in the horizontal grid will start from a depth-grid point with index (M,N).

Starting position in the vertical grid:



A point with index k in the vertical grid will start at the upper layer interface of the grid cell with index k+1 (or the bottom of the grid cell with index k). Consequently, the point with index kmax in the vertical grid will start from the bottom of the grid cell with index kmax, i.e. the sea bottom.

In case the starting position of a point would be a land point and one or more of the surrounding points would be a wet point, it is dis-placed a small distance in the direction of the wet point in order to enable the point to move with the flow from its starting position. If all the surrounding points are situated on land the before mentioned point will evidently not move.

Interpolation of the velocities in horizontal direction It has already been mentioned that, in order to obtain values for the various velocity components at the actual positions of the released points, it is necessary to interpolate using the velocities at the grid points.

For the u- and v-components of the velocity a linear interpolation is used. This rather simple interpolation technique was chosen after it appeared that the more advanced bilinear interpolation led to a too large number of cases to distinguish in the neighbourhood of the coast.

So, if a point is present in the grid cell with index (M,N) this leads to the following result:

$uint = fx \cdot u(M,N) + (1 - fx) \cdot u(M-1,N)$

$vint = fy \cdot v(M,N) + (1 - fy) \cdot v(M,N-1)$

where fx, fy is relative position of a point in a grid cell:

$0 < fx \leq 1$
$0 < fy \leq 1$

The vertical velocity component is not interpolated horizontally.

Interpolation of the velocity components in vertical direction Both the u- and the v-component of the velocity are interpolated quadratically in the vertical direction. This is done to assure smooth transitions between the different model-layers. However not differentiable, the vertical velocity profile in the quadratic case is smooth (compared to a linear interpolation) in the middle of the layers where the quadratic shape functions meet.

This quadratic interpolation has the special property that it doesn't change the vertically averaged velocities. The vertically integrated (layer)velocities before and after interpolation are equal so interpolation does not change the horizontal fluxes. In general this is not true in case a linear interpolation is used.

In case a point is in the upper half part of the top layer or in the lower half part of the bottom layer, no interpolation can take place because only one velocity is available. If a point is found elsewhere in the vertical, we have to distinguish between points in the upper half part of a layer k and points in the lower half part. The formula for the quadratic interpolation used in case a point is in the upper half part of the layer is given by:

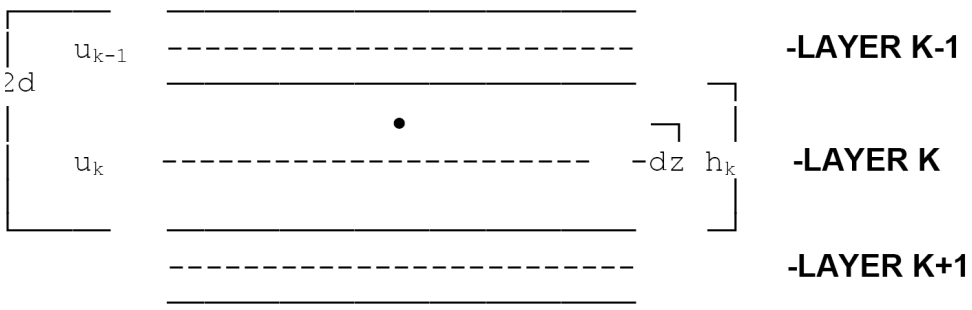
$uint_k = u_k + b \cdot dz + a \cdot dz^2$

with:

$$a = \frac{3}{2d^3}(h_k - h_{k-1}) \cdot (u_{k-1} - u_k)$$

where the following definitions are used:

dz	=	distance from lower situated velocity point
h_k	=	the layer thickness
d	=	0.5 (h_{k-1} + h_k) = the averaged layer thickness of surrounding layers
u_k	=	horizontally interpolated velocity



In case a point has it's starting position in the lower-half part of the layer, k+1 instead of k has to be used in the formulae.

Note that the interpolation becomes linear in case equidistant layers in the vertical are used.

For the w-velocities a simple linear interpolation in the vertical direction is used, because variations in this velocity component are comparatively small.

Integration in time In order to calculate the values for the displacements it is necessary to integrate the velocities in time over the time interval the point was displaced. There is a number of ways to carry out this integration, the simplest of which is probably a forward Euler method. However, a somewhat more sophisticated second order method is implemented in the described option.

A second order time integration is especially recommended in areas with a strongly non-uniform flow pattern to avoid spurious numerical diffusion.

Consider a point which finds itself at a position

x_0 at $t = t_0$, and which was at

x_{-1} at $t = t_0 - \Delta t$

Now the displacement d over a time interval Δt is given by:

$$d(x_0, t_0, \Delta t) = \int_{t_0}^{t_0+\Delta t} u(x_0, t) \cdot dt$$

This displacement will be approximated by the expression:

$d(x_0, t_0, \Delta t) \approx (1.5\ u(x_0, t_0)\ -\ 0.5\ u(x_{-1},\ t_0 - \Delta t)) \cdot \Delta t$

3.4.2.3 Forcing functions

external forces — The values of water level at water level boundaries, velocities at velocity boundaries or discharges at discharge boundaries, are obtained directly from input values. These values, along with wind and discharges provide the external forces that drive the system.

water level and velocity openings

Openings with time-varying water levels, velocities or discharges and Riemann openings are driven by values that are given explicitly for each end of the opening at (regular or irregular) intervals. The values at each point on the opening are obtained by interpolating spatially across the opening and over time to obtain values at each half time step.

For an opening with a distributed total discharge boundary values are given explicitly for the total opening and a special spatial interpolation across the opening is used; a linear interpolation across the opening is used for water level, velocity, discharge and Riemann openings.
The spatial interpolation takes place before the time interpolation.

initial values at time series

In WAQPRE an initial value for a boundary condition at the beginning of a simulation is mandatory.

If the simulation starting time is the same as the first time instance of the time series, the value corresponding to the first time instance will be used. If the starting time of the simulation is not the same as the first time instance of the time series, the starting time will be added to the time series.

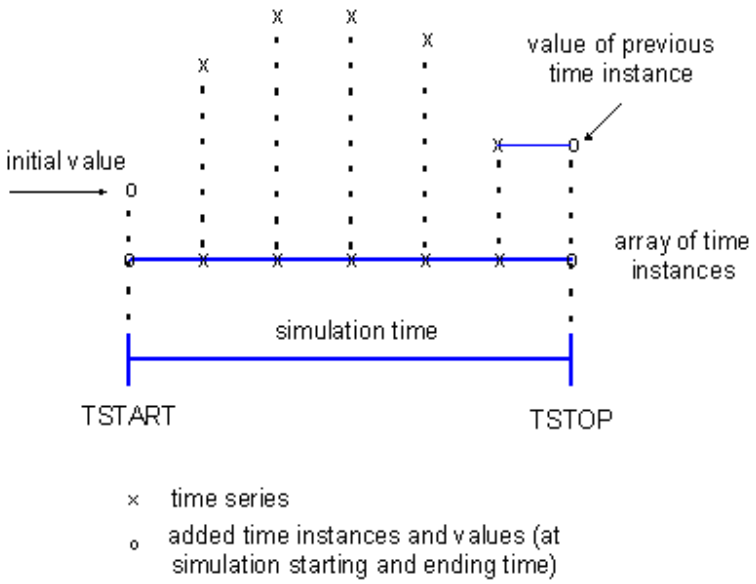


Figure 3.11: Initial value at time series

It is possible that the added simulation starting time is the first time instance of the new time series. In this case the corresponding value of the first time instance will be the specified initial value (see Fig. 3.11).

If the added simulation starting time is not the first time instance of the time series, the value corresponding to the simulation starting time will be calculated by linear interpolation between values corresponding to the previous and the next time instance (see Fig. 3.12).

For the simulation ending time, an analog method is used, except for the situation that the added simulation ending time is the last time instance in the time series. In that case the value corresponding to the previous time instance will be used (see Fig. 3.11).

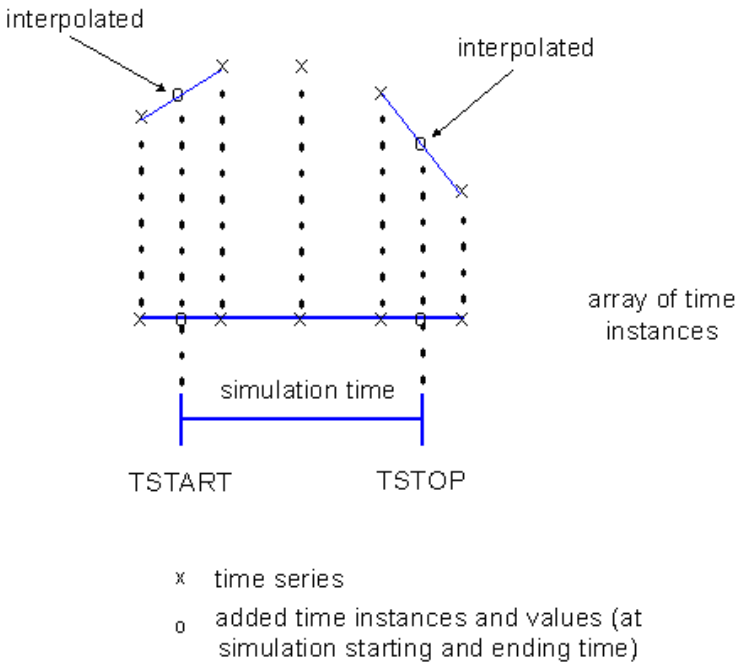


Figure 3.12: linear interpolation

Fourier functions

Openings at which boundary conditions are defined by Fourier functions are driven by values obtained from the equation:

$$x_t = A_0 + \sum_{n=1}^{N} A_n \cos(\omega_n t - \varphi_n)$$

where:

x_t	=	water level or velocity at the open boundary
A_0	=	amplitude at zero frequency
n	=	Fourier component
N	=	number of Fourier components
A_n	=	amplitude
ω_n	=	angular frequency
t	=	elapsed time in seconds (corresponding to an integer times the half time step)
φ_n	=	phase

and A_0, A_n and φ_n are spatially interpolated across the opening.

Riemann openings

Another type of open boundaries are the Riemann invariants. They should however be applied with care, because in the linearisation of the Riemann invariant some assumptions have been made. The most important being that the water level variations (ζ) should be small compared to the water depth (d).
This assumption has the following consequences:

1. The reference plane for the depth should be almost equal to Mean Sea Level.

2. Riemann invariant types of boundary conditions should not be applied in shallow areas where the range of the water level variations is in the order of magnitude of the depth.

For security reasons two examples are given:

1. In an open sea model, in water depth of 2 meter with a tidal range of 1 meter, no Riemann invariants should be applied.

2. In a river model with a reference plane which is far below the water level ranges, no Riemann invariants should be applied.

In WAQUA the following value should be provided:
$u \pm \zeta \sqrt{\frac{g}{d}}$
where:

u	=	velocity at the open boundary
g	=	acceleration by gravity
ζ	=	water elevation
d	=	bottom level (depth)

At lower M and N boundaries the plus sign should be used. At upper M and N boundaries a minus sign should be used.

Distributed discharge openings

A special type of open boundary for WAQUA is a discharge boundary with a spatial distribution of a total discharge through the opening, depending on local water depth and friction value.

The local discharge Q in a point at an opening is computed as:
$Q = H \cdot \Delta y \cdot u$
where:
u = velocity at the open boundary
H = total depth
Δ = grid size at the open boundary
The local velocity u in an opening point is approximated by:
$u \approx C \cdot \sqrt{R \cdot I}$
where:
C = Chezy bottom friction coefficient
R = hydraulic radius
I = hydraulic gradient
The hydraulic radius R in an opening point is approximated by the local depth H. With the assumption of a constant hydraulic gradient I along the opening the ratio between the local discharge Q_i in an opening point i and the total discharge Q_{tot} through an opening can be expressed as follows.

A local weight factor W_i is defined by:

$$W_i = (1-\alpha) \cdot \frac{{H_i}^{3/2} . \Delta\, y_i \,.\, C_i}{\sum_n \left[{H_n}^{3/2} . \Delta\, y_n \,.\, C_n \right]} + \alpha \cdot W_i^{'}$$

where
$W_i^{'}$ = weight factor for opening point i at previous instance
α = relaxation factor
The use of weights of previous times (relaxation) is introduced to be able to damp very fast changes in the discharge distribution in time. The spatial distribution of the user specified total discharge Q_{tot} through an opening is computed with:

$$Q_i = W_i \cdot Q_{tot}$$

Q-H openings	A type of open boundary used for application of WAQUA for river computations is the Q-H relation, specified by means of a table relating total (downstream) discharge and water level. With the computed total discharge through the open boundary the corresponding water is computed from this table (using linear interpolation) and used as a water level boundary condition. For values of the computed total discharge outside the range specified in the Q-H table, the first or last value in the table is used. For time smoothing the computed water levels and user specified initial water levels are used.
Fourier openings	A linear interpolation over time at Fourier openings is used to obtain the water levels and velocities during the interval that simulated time is less than TLSMOOTH. TLSMOOTH is the last time at which interpolation takes place from initial values at open boundaries. It is defined by the user. The interpolation for water levels is between the given initial water level (interpolated across the opening using initial values at both ends) and the computed value (x_t). The interpolation for velocities and discharges is between A0 (interpolated across the opening using phases at zero frequency for both ends) and the computed value (x_t). **Note:** A cosine transform is used in evaluating the Fourier function in WAQPRO.

3.4.2.4 Chezy coefficient computation

The pre-processor WAQPRE computes Chezy coefficients at intervals of TICVAL. TICVAL is the user defined time interval to compute Chezy values from given friction values. The Chezy values are calculated both at u- and v-velocity locations.

The Chezy coefficient for a particular velocity point can be computed by four methods:

1) Manning:

$$C = \frac{\sqrt[6]{H}}{n}$$

where:

C	=	Chezy coefficient ($m^{1/2}$ / s)
H	=	total depth at the velocity point (m)
n	=	Manning's parameter ($m^{-1/3}$ s)

2) White-Colebrook:

$$C = 18 \cdot {}^{10}\log\left(\max\left(\frac{12H}{k_\eta}, 1.0129\right)\right)$$

where:

C	=	Chezy coefficient ($m^{1/2}$ / s)
H	=	total depth at the velocity point (m)
k_η	=	k-Nikuradse value (m)

Reference: Leo C. van Rijn: Principles of fluid flow and surface waves in rivers, estuaries, seas and oceans (pp 82-83).

3) Chezy: no conversion takes place.

Chezy-salinity correction

When transport of constituents is involved in the simulation and one of the constituents is salt, a correction may be applied to the bottom roughness.

The user can control the correction by the input coefficient ALFA_CHEZY, which is α in the formula for the correction factor β described below.

formulation

The momentum equation in the u-direction used in WAQUA is as follows:
$\frac{\partial u}{\partial t} + \ldots + \beta g \frac{\|\vec{u}\| u}{C^2 H} + g\frac{\partial \zeta}{\partial x} = \frac{g}{2}\frac{H}{\rho}\frac{\partial \rho}{\partial x} + \ldots$
In this H is the total water depth, ζ is the water level and ρ is the density. Special is the coefficient β in the friction term which will be described below. The dots (. . .) stand for terms that are not of interest, such as advective terms, Coriolis, wind and viscosity.

Due to the effect that a strict horizontal pressure gradient generates velocities at the bottom different from the vertically integrated velocity, the bottom friction term must be adapted with a factor β. It is assumed that the velocity at the bottom has the same direction as the vertically integrated velocity.

For β applies:
$$\beta = 1 - \alpha \frac{H}{\rho} \frac{\vec{\nabla}\rho \cdot \vec{u}}{\|\vec{u}\|}$$
α is a dimensionless coefficient given by the user. β must have a value between $\frac{1}{3}$ and 3.

The adaptation is done with the Chezy values. If C stands for the not corrected Chezy value and C' for the corrected one, then:
$$\frac{1}{C'^2} = \frac{\beta}{C^2} \Leftrightarrow C' = \frac{C}{\sqrt{\beta}}$$
In a Cartesian grid the inner product of the gradient of ρ and the velocity vector reads:
$$\vec{\nabla}\rho \cdot \vec{u} = \frac{\partial \rho}{\partial x} u + \frac{\partial \rho}{\partial y} v$$
In an orthogonal curvilinear grid the inner product is:
$$\vec{\nabla}\rho \cdot \vec{u} = \frac{1}{\sqrt{g_{\xi\xi}}}\frac{\partial \rho}{\partial \xi} u + \frac{1}{\sqrt{g_{\eta\eta}}}\frac{\partial \rho}{\partial \eta} v$$
In the latter formula, u and v are the curvilinear velocity components.

4) Linear bottom friction is given by:
$$H\frac{\partial u}{\partial t} + \lambda u + ... = ...$$
which results in the following Chezy coefficient:
$$C = \frac{\sqrt{g}(u^2 + v^2)^{1/4}}{\sqrt{\lambda}}$$
where:

C	=	Chezy coefficient ($m^{1/2}$ / s)
H	=	total depth at the velocity point (m)
g	=	Gravity accelleration (m/s^2)
λ	=	Linear fraction factor

3.4.2.5 Nikuradse roughness

General

The Nikuradse option is only available in combination with the White-Colebrook roughness method.

When computations are made with the NIKURADSE option, Chezy roughness will be calculated from the field characteristics as described by the AREA-U and AREA-V files. The roughness at each velocity point is calculated as a summation of all roughness elements proportional to their relative surface area or relative length (both defined by the area files).

Nikuradse k-values

Nikuradse k-values and area's in u and v - excluding lines - are summed as follows:

$$k_{u,cell} = \sum_{r_code=1}^{950} (k_{r_code,cell} \cdot area_u_{r_code,cell})$$

$$k_{v,cell} = \sum_{r_code=1}^{950} (k_{r_code,cell} \cdot area_v_{r_code,cell})$$

where the roughness code r_code may occur several times within one cell. The summations are made for the area that is not occupied by buildings. If buildings are present within a grid cell a correction is made for all area's ($area_u_{r_code,cell}$). No account is taken for the position of the areas within the grid cell.

hedges

Next, line structures like hedges are added. The Chezy values of all hedges in one cell are summed according to:

$$C_{u,cell} = \sqrt{\frac{1}{\sum_{r_code=951}^{999} \left(\frac{1}{C^2_{r_code,u,cell}} \right)}}$$

$$C_{v,cell} = \sqrt{\frac{1}{\sum_{r_code=951}^{999} \left(\frac{1}{C^2_{r_code,v,cell}} \right)}}$$

where the roughness code r_code may occur several times within one cell.

Corrections are made for the area occupied by buildings. Account, for the direction of the hedge, is taken by projection of the space occupied by the hedge upon the gridlines. No account is taken for possible sheltering of hedges by buildings, vegetation structures or hedges in the same grid cell. The Chezy values of all hedges in the cell are added tot the k value using:

$$k_{total,u,cell} = k_{u,cell} + \frac{12 \cdot h}{10^{\frac{C_{u,cell}}{18}}}$$

$$k_{total,v,cell} = k_{v,cell} + \frac{12 \cdot h}{10^{\frac{C_{v,cell}}{18}}}$$

Static roughness

static	For roughness codes (R_CODE) up to 400 Nikuradse values are obtained directly from the roughness table. See User's Guide WAQPRE section 2.8.1.5 NIKURADSE.

Alluvial roughness

alluvial

For roughness codes (R_CODE) from 401 until 700 (usually projected in the main channels of rivers) Nikuradse-values are calculated using the following equation:

$$k_{alluvial} = \alpha_{alluvial} \cdot h^{0,7} \cdot \left(1 - e^{-\beta \cdot h^{-0.3}}\right)$$

where

$k_{alluvial}$	=	Nikuradse k (m)
$\alpha_{alluvial}$	=	coefficient for calibration
β	=	coefficient for calibration
h	=	waterdepth (m)

This equation was, through several neglections, derived from the bedroughness predictor of Van Rijn (RWS-RIZA, publ. Rijn 1997-42 H: L.C.van Rijn, Principles of Sediment transport, chapter 6). Since bedroughness predictions are of low accuracy, calibration of bedroughness with $\alpha_{alluvial}$ is considered essential for proper hydraulic modelling.

Roughness of vegetation structures types

vegetation

When roughness codes lie in between 700 and 951 (700 and 951 excluded) roughness is calculated for flow through vegetation or flow through and over vegetation. For this computation, the following equation described by Barneveld H.J. (et.al. HKV, pr051) is used for each velocity point in the computational field.

$$C = \chi \cdot \frac{1}{h^{1}/_{2}} \cdot \left\{ \begin{array}{l} \frac{2}{\sqrt{2A}} \cdot (\sqrt{C_3\, e^{k\sqrt{2A}} + u_{v0}^2} - \sqrt{C_3 + u_{v0}^2}) + \\ \frac{u_{v0}}{\sqrt{2A}} \cdot \ln\left(\frac{(\sqrt{C_3\, e^{k\sqrt{2A}} + u_{v0}^2} - u_{v0}) \cdot (\sqrt{C_3 + u_{v0}^2} + u_{v0})}{(\sqrt{C_3\, e^{k\sqrt{2A}} + u_{v0}^2} + u_{v0}) \cdot (\sqrt{C_3 + u_{v0}^2} - u_{v0})}\right) + \\ \frac{\sqrt{g(h-(k-a))}}{\kappa} \cdot \left((h - (k-a)) \ln\left(\frac{h-(k-a)}{z_0}\right) - a \ln\left(\frac{a}{z_0}\right) - (h-k)\right) \end{array} \right\}$$

The variables a, α, C_3, z_0 and the constants A, E_1, F, u_{v0} are calculated according to the following equations:

$$
\begin{aligned}
A &= \frac{m \cdot D \cdot C_D}{2\alpha} \\
a &= \frac{1 + \sqrt{1 + \dfrac{4 \cdot E_1^2 \cdot \kappa^2 \cdot (h-k)}{g}}}{\dfrac{2 \cdot E_1^2 \cdot \kappa^2}{g}} \\
\alpha &= 0.01\sqrt{h \cdot k} \text{ and } \alpha \geq\ 0.001 \\
C_3 &= \frac{2 \cdot g \cdot (h-k)}{\alpha \cdot \sqrt{2 \cdot A} \cdot (e^{k \cdot \sqrt{2 \cdot A}} + e^{-k \cdot \sqrt{2 \cdot A}})} \\
E_1 &= \frac{\sqrt{2 \cdot A} \cdot C_3 \cdot e^{k \cdot \sqrt{2 \cdot A}}}{2 \cdot \sqrt{C_3 \cdot e^{k \cdot \sqrt{2 \cdot A}} + u_{v0}^2}} \\
F &= \frac{\kappa \cdot \sqrt{C_3 \cdot e^{k \cdot \sqrt{2 \cdot A}} + u_{v0}^2}}{\sqrt{g \cdot (h - (k-a))}} \\
u_{v0} &= \sqrt{\frac{2 \cdot g}{C_D \cdot m \cdot D}} \\
z_0 &= a \cdot e^{-F}
\end{aligned}
$$

where

a = distance separation layer from surface (m)
α = characteristic length of large eddies (m)
C = Chezy roughness coefficient ($m^{0.5} \cdot s^{-1}$)
C_d = drag coefficient; a constant value of 1.65 is used.
C_3 = constant
χ = calibration parameter
D = average stem diameter (m)
E_1 = constant
F = constant
g = acceleration of gravity ($m \cdot s^{-2}$)
h = water depth (m)
k = vegetation height (m)
κ = Von Karman constant
m = number of stems per square meter horizontal surface area
u = velocity ($m \cdot s^{-1}$)
u_{s0} = characteristic velocity in vegetation layer ($m \cdot s^{-1}$)
u_{v0} = intermediary constant for velocity in vegetation layer ($m \cdot s^{-1}$)
z = distance in vertical (m)
z_0 = virtual bottom roughness of surface layer (m)

If the vegetation height is larger than the water depth the equation for C can be simplified to:

$$C = \chi \sqrt{\frac{2g}{C_d \cdot m \cdot D \cdot h}}$$

In all cases it is assumed that the vegetation density does not vary in the vertical below the canopy (elevation < k). The vegetation density $m \cdot D$ is given in kolom B of the Rcode-table of User's guide WAQPRE, section 2.8.1.5 NIKURADSE. In addition, the effects of different flow profiles (velocity distribution in vertical) on resistance were neglected. Finally it should be noted that it may be difficult to obtain accurate values for m, D, k and β may not be too accurate. Considering this, we strongly recommend calibration of C with χ for field conditions.

Hedges

hedges

When r_codes lie between 950 and 1000 (both excluded) roughness of hedges are calculated. Hedge height, density and projected relative length are read from the U or V file. Hedges are, unlike lanes, interpreted as line structures. The hedges are projected upon the gridlines through the velocity points.
First of all the hedge passfactor is calculated as follows.

If the hedge is not submerged (k≥h) by:
$\mu_{pass} = 1 + 0.175\, n\, \left(\frac{h}{k} - 2\right)$
If the hedge is submerged (k<h) by:
$\mu_{pass} = 1 - 0.175\, n\, \left(\frac{k}{h}\right)$
And the passfactor is limited to the following range:
$0.001 \leq \mu_{pass} \geq 0.999$
When the passfactor is defined, the Chezy value for the hedge is calculated using:

$$C_{hedge} = \sqrt{\frac{2g \cdot l_{grid} \cdot \mu_{pass}^2}{r_{project} \cdot h \cdot (1 - \mu_{pass}^2)}}$$

where

h	=	water depth (m)
k	=	hedge height (m)
l_{grid}	=	grid length belonging to velocity point (m)
n	=	number of stems per meter hedge
$r_{project}$	=	projection length hedge relative to grid length (m)
μ_{pass}	=	hedge passfactor

So far no calibration parameter for hedges has been introduced.

3.4.2.6 Roughcombination roughness

General

The Roughcombination option is an improved Nikuradse method with several extensions. This method can combine several roughness methods, even within one cell. The method is available in combination with all the possibilities given in FORMULA except the Z0-based method. The chosen method in FORMULA will be used as the default method. The user can overrule the default method at each cell by using the area-U and area-V files. When computations are made with the ROUGHCOMBINATION option, Chezy roughness will be calculated from the field characteristics as described by the AREA-U and AREA-V files. The roughness at each velocity point is calculated as a summation of all roughness elements in one grid cell proportional to their relative surface area or relative length in that grid cell (both defined by the area files). In the summation a serial and a parallel chezy computation is used and finally combined with a weighting factor theta into an overall chezy value per grid cell.

For general information about the vegetation options and formula's see the documents
"Stromingsweerstand vegetatie in uiterwaarden, Deel Handboek versie 1-2003"

RIZA rapport 2003.028 and "Stromingsweerstand vegetatie in uiterwaarden, Deel 2 Achtergronddocument versie 1-2003" RIZA rapport 2003.029 by E.H. van Velzen, P. Jesse, P. Cornelissen and H. Coops. For information about the buildings see the document "Bepaling van de gemiddelde invloed van gebouwen op de ruwheid" by Ubels. For general information about the input description see User's Guide WAQPRE section 2.8.1.5 ROUGHCOMBINATION

White-Colebrook

White-Colebrook

For roughness codes (R_CODE) from 101 until 300 (both included) Nikuradse values are obtained directly from the roughness table. The chezy coefficient is computed with the White-Colebrook formula.

$$C = 18 \log \{ \max (\frac{12H}{k_n}, 1.001) \}$$

where:

C	=	Chezy coefficient ($m^{1/2}$ / s)
H	=	total depth at the velocity point (m)
k_n	=	k-Nikuradse value (m)

Manning

Manning

For roughness codes (R_CODE) from 301 until 500 (both included) Manning values are obtained directly from the roughness table. The chezy coefficient is computed with the Manning formula:

$$C = \frac{\sqrt[6]{H}}{n}$$

where:

C	=	Chezy coefficient ($m^{1/2}$ / s)
H	=	total depth at the velocity point (m)
n	=	Manning's parameter ($m^{-1/3}$ s)

Chezy

Chezy

For roughness codes (R_CODE) from 501 until 600 (both included) Chezy values are obtained directly from the roughness table. No conversion takes place.

Alluvial roughness

alluvial

For roughness codes (R_CODE) from 601 until 900 (usually projected in the main channels of rivers) Nikuradse-values are calculated using the following equation:

$$k_{alluvial} = \alpha_{alluvial} \cdot h^{0,7} \cdot \left(1 - e^{-\beta \cdot h^{-0.3}}\right)$$

where

$k_{alluvial}$	=	Nikuradse k (m)
$\alpha_{alluvial}$	=	coefficient for calibration
β	=	coefficient for calibration
h	=	waterdepth (m)

This equation was, through several neglections, derived from the bedroughness predictor of Van Rijn (RWS-RIZA, publ. Rijn 1997-42 H: L.C.van Rijn, Principles of Sediment transport, chapter 6).
Since bedroughness predictions are of low accuracy, calibration of bedroughness with $\alpha_{alluvial}$ is considered essential for proper hydraulic modelling. Because β is not very sensitive in the formula the default value 2.5 is not often changed.

Roughness of vegetation structures types (area's)

vegetation

For roughness codes (R_CODE) between 1201 and 1400 (both included) roughness is calculated for flow through vegetation or flow through and over vegetation. The following formula is used

For waterdepths greater then vegetation heights (h > k):

$$C_r = \frac{k.U_v + (h-k).U_o}{h\sqrt{h.i}}$$

with:

$$U_v = \frac{2}{k\sqrt{2A}}(\sqrt{C_2.e^{k\sqrt{2A}} + u_{s0}^2} - \sqrt{C_2 + u_{s0}^2}) + \frac{u_{s0}}{k\sqrt{2A}}.\ln\left[\frac{(\sqrt{C_2.e^{k\sqrt{2A}} + u_{s0}^2} - u_{s0})(\sqrt{C_2 + u_{s0}^2} + u_{s0})}{\sqrt{C_2.e^{k\sqrt{2A}} + u_{s0}^2} + u_{s0})(\sqrt{C_2 + u_{s0}^2} - u_{s0}}\right]$$

with:

$$U_o = \frac{u_*}{\kappa\,(h-k)}\left[(h-(k-a)).\ln(\frac{h-(k-a)}{z_0}) - a.\ln(\frac{a}{z_0}) - (h-k)\right]$$

with:

$$\alpha = 0.0227.k^{0.7}$$

$$A = \frac{C_d.A_r}{2\alpha}$$

$$B = -\frac{gi}{\alpha}$$

$$u_{s0}^2 = \frac{k.i}{\frac{A_r * k * C_d}{2g} + \frac{1}{C_b^2}}$$

C_b is defined as:

$$C_b = 18\log(\frac{12h}{k_b})$$

$$C_1 = \frac{+2B(h-k)}{\sqrt{2A}.(e^{k\sqrt{2A}} + e^{-k\sqrt{2A}})}$$

$$C_2 = -C_1$$

$$E = \frac{\sqrt{2A}(-C_1 e^{-k\sqrt{2A}} + C_2.e^{k\sqrt{2A}})}{2\sqrt{C_1 e^{-k\sqrt{2A}} + C_2.e^{k\sqrt{2A}} + u_{s0}^2}}$$

$$F = \frac{\kappa.\sqrt{C_1 e^{-k\sqrt{2A}} + C_2.e^{k\sqrt{2A}} + u_{s0}^2}}{\sqrt{g(h-(k-a)).i}}$$

$$a = \frac{1 + \sqrt{1 + \frac{4.E^2.\kappa^2.(h-k)}{g.i}}}{\frac{2.E^2.\kappa^2}{gi}}$$

$z_0 = a.e^{-F}$

$u_* = \sqrt{g(h-(k-a)).i}$

For waterdepths less then or equal on the vegetation heights ($h \leq k$)

$$C_r = \sqrt{\frac{1}{\frac{A_r.h.C_d}{2g} + \frac{1}{C_b^2}}}$$

where:

C_r	=	representative Chezy coefficient $[m^{1/2}/s]$
k_r	=	representative Nikuradse sand roughness $[m]$
h	=	water depth $[m]$
k	=	vegetation height $[m]$
A_r	=	vegetation density $[m^2/m^2/m']$
C_d	=	drag coefficient [-]
k_b	=	Nikuradse sand roughness bottom $[m]$
C_b	=	Chezy coefficient for bottom roughness $[m^{1/2}/s]$
i	=	slope of the waterlevel [-]
α	=	characteristic length of large eddies $[m]$
g	=	acceleration of gravity 9.81 $[m/s^2]$
κ	=	von Karman constant : 0.4 [-]
u_{s0}	=	characteristic velocity in vegetation layer $[m/s]$
u_*	=	shear stress at boundary of vegetation/water layer $[kgm^{-1}s^{-2}]$
z_0	=	virtual bottomroughness waterlayer $[m]$
a	=	distance separation layer from surface $[m]$
U_0	=	average stream velocity (vertical) in the water layer $[m/s]$
U_v	=	average stream velocity (vertical) in the vegetation layer $[m/s]$
A,B	=	work variable
C_1,C_2	=	work variable
E,F	=	work variable

Buildings or water free surfaces

buildings

For roughness codes (R_CODE) from 1 until 50 (both included) Chezy values for buildings are calculated as follows:
$C_{withbuildings} = C_{withoutbuildings} \cdot (1.12 - 0.25B - 0.99\sqrt{B})$
where:
B = area with buildings in this grid cell / area of the grid cell
The valid values of B are between 0.014 and 0.843. The program will fit the values outside the valid range to this minimum and maximum.

The formula is developed for buildings, but it is also used for water free surfaces.

Roughness of vegetation structures types (points, individual trees)

individual trees

For roughness codes (R_CODE) between 1501 and 1600 (both included) roughness of individual trees are calculated. The vegetation density is read from the area-U or area-V file. The Chezy value for the individual trees is calculated using:

If the tree is not submerged (k≥h) by:

$$C_r = \sqrt{\frac{1}{\frac{1}{C_{basis}^2} + \frac{C_d.D.k.N}{2.g.OR}}}$$

If the tree is submerged (k<h) by:

$$C_r = \sqrt{\frac{1}{\frac{1}{C_{basis}^2} + \frac{C_d.D.h.N}{2.g.OR}}}$$

with:

C_{basis}	=	Chezy coefficient base roughness $[m^{1/2}/s]$
C_d	=	drag coefficient [-]
D	=	stem diameter $[m]$
k	=	height of the tree $[m]$
h	=	waterdepth $[m]$
N	=	number of trees within the grid cell [-]
OR	=	area of the grid cell $[m^2]$
C_r	=	representative Chezy value for base roughness
	+	individual trees $[m^{1/2}/s]$

Roughness of vegetation structures types (lines, hedges)

hedges

For roughness codes (R_CODE) between 1601 and 1700 (both included) roughness of hedges are calculated. The projected relative length is read from the area-U or area-V file. Hedges are regarded as line structures. The hedges are horizontal perpendicular projected upon the gridlines through the velocity points. The Chezy value for the hedge is calculated using:

If the hedge is not submerged (k≥h) by:

$$C_h = \sqrt{\frac{2g.a}{h}} \cdot \left[\frac{k}{h} \cdot \sqrt{\frac{1}{C_d.A_h}} \right]$$

If the hedge is submerged (k<h) by:

$$C_h = \sqrt{\frac{2g.a}{h}} \cdot \left[\frac{k}{h} \cdot \sqrt{\frac{1}{C_d.A_h}} + m_o \cdot \sqrt{\frac{\left(\frac{h-k}{h}\right)^2}{1-\left(\frac{h-k}{h}\right)^2}} \right]$$

where

C_h	=	Chezy coefficient hedge [$m^{1/2}$/s]
C_d	=	drag coefficient [-]
A_h	=	hedge density [m/m]
k	=	height of the hedge [m]
a	=	distance between hedges (grid cell distance) [m]
h	=	waterdepth [m]
g	=	acceleration gravity 9.81 [m/s^2]
m_0	=	energy loss coeffiecient [-]

If there is more then one hedge in a grid cell the total chezy value of the hedge is calculated as follows:

$$\frac{1}{C_r^2} = \frac{1}{C_{h1}^2} + \frac{1}{C_{h2}^2}$$

where:

C_r	=	Resulting Chezy value for all hedges [$m^{1/2}$/s]
C_{h1}	=	Chezy value for hedge one [$m^{1/2}$/s]
C_{h2}	=	Chezy value for hedge two [$m^{1/2}$/s]

Summation of roughness values

Summation roughness values

All roughness methods (including line information like hedges and point information like individual trees) are translated to Chezy values. If there is more then one Chezy value per grid cell, due to more then one roughness method per grid cell, a overall Chezy value is calculated with the following formula:
$C_{rc} = \phi.C_s + (1-\phi).C_p$
where:
$\phi = 0.6$
$C_p = \sum_i x_i.C_{ri}$
and
$$C_s = \frac{1}{\sqrt{\sum_i \frac{x_i}{C_{ri}^2}}}$$
with :

x_i	=	area fraction roughness type i [-]
C_{ri}	=	Chezy value roughness type i [$m^{1/2}$/s]
C_p	=	Chezy value for parallel pattern [$m^{1/2}$/s]
C_s	=	Chezy value for serial pattern [$m^{1/2}$/s]
C_{rc}	=	Representative Chezy value for combination of different roughness types [$m^{1/2}$/s]
ϕ	=	weighting factor [-]

where the roughness code C_{ri} may occur several times within one cell. The summations are made for the area that is not occupied by buildings. If buildings are present within a grid cell a correction is made for all area's. No account is taken for the position of the areas within the grid cell.

3.4.2.7 Equation of state

density

In the case where the computations are made in the baroclinic mode, the density has to be computed from the salinity. For this computation, the following equation of state described by Eckert is used for each point of the computational field:

$$\rho = \frac{1000\ (5890 + 38\mathrm{T} - 0.375\,\mathrm{T}^{\,2} + 3\mathrm{s})}{(1779.5 + 11.25\,\mathrm{T} - 0.0745\,\mathrm{T}^{\,2}) - (3.8 + 0.01\,\mathrm{T}\,)\,\mathrm{s} + \alpha_0\left(5890 + 38\,\mathrm{T} - 0.375\ \mathrm{T}^{\,2} + 3\mathrm{s}\right)}$$

where:

ρ	=	density
s	=	salinity (g/l) (for sea water approximately 30 g/l, kg/m^3)
T	=	temperature (centigrade)
α_0	=	constant. Eckert uses 0.698.

The density is computed by this equation in kg/m^3. This is inconsistent with the units used in the simulation, but by expressing the reference density RHOREF which appears at the same time also in kg/m^3, this inconsistency has no effect on the simulation results.

Reference: C. Eckert: The Equation of State of Water and Sea Water at Low Temperatures and pressures Am. Jour. of Sci. Vol. 256, 1958, pp. 240-250.

3.4.2.8 Non-hydrostatic computations

TRIWAQ has the possibility to take non-hydrostatic effects for free-surface flows into account. In this method vertical velocities are solved from the momentum equation in z-direction. Every time step the flow is made divergence free with a pressure-correction method in which a Poisson equation is solved for the full three-dimensional field.

Reference: M. Zijlema and G.S. Stelling: Further experiences with computing non-hydrostatic free-surface flows involving water waves. Int. J. Numer. Meth. Fluids. Vol. 48, 2004, pp. 169-197.

3.5 Special hydrodynamical objects

3.5.1 Barriers and sluices

3.5.1.1 Features

time varying constriction
: The modeller may define certain grid points to be sluices or barriers, which are constructions with time-varying constriction on velocity in the u-direction or the v-direction. Constriction on both components of velocity is effected by two barriers at the same grid point. In the vertical, constriction of the flow varies both by the raising and lowering of a sill and by the lowering and raising of a gate. In the horizontal, effective width may be increased or decreased for more or less constriction. u-barriers are located on u-velocity points and v-barriers on v-velocity points. The length of the barrier is assumed to be zero.

(sub)critical flow
: Since the constriction may cause supercritical flow, both subcritical and supercritical conditions are taken into account in the WAQUA computation.

  Note: It should be noted, however, that in TRIWAQ only (gate-restricting) subcritical flow can be considered for the moment!

application
: The description of barriers which is given below, is originally based on Leendertse's books. Leendertse however situates a barrier at the water level point. That way of applying barriers was debated and improved (see 'Aspecten Barriers WAQUA-modellen', A. Langerak, DGW notitie GWAO 88.808, 'Schematisatie van barriers in WAQUA', J. Wijbenga, WL rapport Q779, may 1989 and 'Niet-aansluitende overgangen tussen verschillende toestanden van barriers in WAQUA', dr.ir. E.A.H. Vollebregt, technical report TR05-03, VORtech Computing, 2005 and 'Vernieuwing kunstwerkformulering in WAQUA', eindrapport ontwikkeling prototype, technisch rapport BvP/1383/6697, 1 december 2006). In WAQUA barriers are situated at the velocity points. It is possible to define more than one barrier in a row or column. Diagonal barriers are also implemented (not in TRIWAQ).

energy loss
: At a barrier, the flow is restricted in some form and a special computation is made to allow for the energy loss at the constriction besides of the bottom friction loss.
  Furthermore, the flow will be restricted when the barrier transits into a supercritical condition. By this, the flow rate through the barrier cannot exceed the critical flow rate.

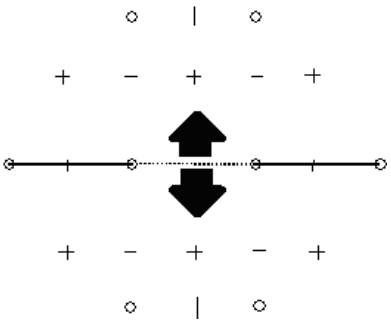


Figure 3.13: Typical v-barrier between closed v-velocity points

location

A barrier is located at a velocity point and has to be specified to be a u-flow or v-flow barrier. A u-flow and a v-flow barrier can have the same (m, n) coordinates. Each of them works independently of the other in this case. Such a layout is shown in Fig. 3.14. Typically a barrier is located between dam points, closed velocity points and/or other barrier points (Fig. 3.13). For the restrictions on location of barriers see the User's Guide WAQPRE.

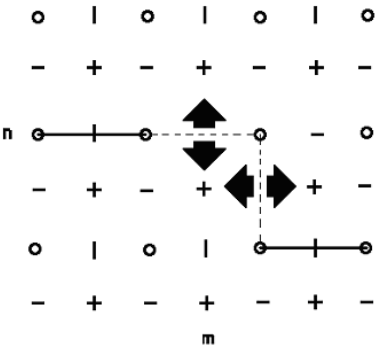


Figure 3.14: Barriers in two flow directions at location m,n

dynamic effects

When we are dealing with non-steady flow, dynamic effects near a barrier are accounted for in the equation of motion. However, the pressure differences resulting from density gradients are not taken into consideration.

monitoring

Since the water levels at each side of a barrier and the flow rates and velocities are generally of much practical interest, these variables are always printed when water levels, currents, and flows through cross-sections are requested. All this data is also written on the SDS file and time histories of many variables at or near the barrier can be plotted. Also the barrier flow condition (see further on), the estimated flow-through height and velocity, and the energy loss are stored. There is no need to insert water level and current stations near the barrier for this purpose.

time variation

The sill depth, the gate height and the width/grid size ratio can be varied in time, which makes it possible to make a simulation for solving practical engineering problems.

time interpolation

Interpolation across time occurs for time series of the three time-varying barrier dimensions just mentioned. The time series values can be equidistant or not. The value at the beginning of interpolation is either the previous value given in the input, or the initial value (given or assumed) (see paragraph 3.4.2.3).

line barrier

It is also possible to define a barrier to be located along a line. A line barrier can only be defined along a grid line (thus the M-coordinate or the N-coordinate must be constant).

During the computation line barriers will be converted into point barriers. The print output will give the information at grid points.

In general line barriers act in the same way as point barrier. There is however one main difference with point barriers, namely the barrier width. Gate and sill level for all barrier points on a line barrier will be the same, but not the barrier width. A partially closed barrier (width < 1.0) will be closed from the edges of the line.
Example 1:

Suppose a line barrier is defined by:

```
Points
    P1 = (m=1, n=1)
    P2 = (m=5, n=1)
Curves
    C1 = Line(P1,P2)
Barriers
    B1:  C1
```

This means the line barrier B1 consists of 5 barrier points if the barrier width is set to 0.5 (that is 50 % of the barrier is open). In the rectangular case the individual barrier widths are set to:

Barrier nr.	M	N	Width
1	1	1	0.0
2	2	1	0.75
3	3	1	1
4	4	1	0.75
5	5	1	0.0

Example 2:

Same as above, but now for the curvilinear case. The barrier widths will now be:

Barrier nr.	Grid space	M	N	Width
1	400.0	1	1	0.375
2	300.0	2	1	1
3	150.0	3	1	0.333
4	100.0	4	1	0
5	50.0	5	1	0

In the above example the physical width of the barrier is 1000 meter. When this barrier is closed for 50 barrier must be closed.
As a result barrier point (1,1) will be partially closed. The barrier width will be set to 150/400 = 0.375.
Barrier points (4,1) and (5,1) will be fully closed this makes 150 m. Barrier (3,1) must be closed partially (100 m.) in order to end up to 250 m. The barrier width will be set to 50/150.0 = 0.333.

barrier steering

During the computation it is possible to change the dimensions of the barrier (sill level, gate level, width). There are two direct ways to accomplish this: first by means of a time series, second by means of a table.

In real life the rate of change in the position of parts of a movable barrier is usually limited. For this case it is possible to define maximum velocities for the GATE, SILL and WIDTH.

During the computation the steering of a barrier can be changed when some condition is met. When a condition is triggered a new timeseries or table may be activated for steering the barrier.

Finally the start time, end time and time interval for changing the barrier dimensions can be calculated. At default the barrier dimensions will be calculated at all half time steps.

time series

Time series can be defined in two ways. It is important to understand the differences between both options.

The first option is the oldest one and is primarily available for reasons of compatibility. The time series are defined directly under keyword FLOW - FORCINGS - BARRIERS and are given for each dimension (SILL, GATE and WIDTH) separately.

Note: In this case the time that is given for a time series is relative to Starting date of the simulation at midnight (0:00).

The second option is to define a time series under keyword FLOW - FORCINGS - BARSERIES, and refer to this time series in the GLOBAL part of the description of a barrier under keyword FLOW - FORCINGS - BARRIERS. In this case it is possible to define a time series for all moveable parts of the barrier at once. This might be all dimensions, but also only e.g. the gate level.

Note: In this case the time that is given for a time series is relative to the moment the time series is activated. This can be the starting time of the simulation, or the moment that a condition is triggered (see below).

tables

A time series gives a fixed steering strategy. By using a table the user can have the barrier dimensions vary depending of some parameter. The possible parameters are:

- Water level or concentration at a point.
- Discharge over a cross section.
- Water level, water pressure or concentration difference between two points, or difference between a point and an observed value.
- Discharge difference between two cross sections or between a cross section and an observed value.

Example:

The next barrier table is linked to the water level in a point

T1: values=

#Sill	Gate	Width	Parameter
1.0	5	1.0	0
0.	5	1.0	1
0.	5	0.0	1.1

The next table shows what happens to the dimension if the water level (i.e. the parameter) varies:

Water level	Sill	Gate	Width
-1.3	1.0	5.0	1.0
0.3	0.7	5.0	1.0
1.0	0.0	5.0	1.0
1.2	0.0	5.0	0.0

maximum velocities

The previous example shows, when using a table (but also time series), a barrier might be closed from one instance to another. This is not always wanted for stability reasons but also because of physical restrictions of the barrier. For a large barrier it will take some time to close or open.

For that reason the variation of the different dimensions can be limited by defining a maximum velocity for the dimensions. The maximum velocity can be defined to be an absolute velocity or a relative velocity to the actual opening. In case of a relative velocity, a barrier that is initially closed will not open, and a barrier that should be fully closed, will not completely close. For that reason a minimal velocity can be specified.

When using a relative velocity the maximum velocity is set by the following formula:

MaxVel = max (RelVel * Opening, MinVel)

Where:

MaxVel	=	Maximum velocity for changing barrier dimensions
RelVel	=	Specified relative velocity
Opening	=	Actual opening. For the sill level the opening will be equal to the distance between the minimum of the water level and gate level, and the sill level. For the gate level the opening will be equal to the distance between the gate and the sill. For the width the opening will be equal to the actual width.
MinVel	=	Specified minimum velocity.

conditions

The steering mechanism of a barrier can be changed using "conditions".

In the conditions about the same parameters can be used as in the barrier tables, namely:

- Wind speed or direction.
- Water level or concentration at a point.
- Discharge over a cross section.
- Water level, water pressure or concentration difference between two points, or difference between a point and an observed value.
- Discharge difference between two cross sections or between a cross section and an observed value.
- The dimensions of a barrier: sill depth, gate height and relative barrier width.

Example:

This example shows a possible condition for a barrier:

Condition

if (Discharge C2 gt 0.0) and (Level P1 gt 0.5) and

(Sill B1 gt 4.49) then TS1

elseif (Level P3, P2 lt 0.0) and (Sill B1 lt -2.99) then TS2

endif

In the above example the steering of the barrier will start using the first time series (TS1) when the first condition becomes true, i.e. when the discharge through C2 is positive, the waterlevel in checkpoint P1 is larger than 0.5m, and the sill of barrier B1 is 4.5m or more below the reference plane (assuming that sill-values are given positive-downwards). The starting time of the time series is set to 0. If the first condition is still true when the conditions are evaluated later on, no action will be taken.

If the first condition is (becomes) false then the second condition will be taken into consideration. The condition "waterlevel P3, P2 lt 0.0" means:

water level in point P3 - water level in point P2 lt 0.0

If the condition evaluates to true, time series TS2 will be activated. Instead of a time series one may also activate a table.

When both conditions in the example evaluate to false, no action will be taken. This means that the table or time series that is in use will be continued. Note that this behaviour may be changed by adding an else-clause. In that case the table or time series that is named after the keyword 'else' is activated as soon as all the conditions become false. This is not desired in the current example, where TS1 describes closing of the barrier (may only be started when the barrier is completely open, sill=4.5), TS2 describes opening of the barrier (may be started only when the barrier is completely closed, sill=-3.0), and where opening and closing of the barrier are both processes that must be completed once they have been started.

barrrier steering program

For each barrier the barrier steering program will execute the following steps:

1. If a condition is given for the barrier, then the program checks if the steering of the barrier has to be changed, i.e. whether a new time series or table is activated.
2. Depending of the actual steering of the barrier (time series or a table), the preferred dimensions (given by the active time series or table) of a barrier will be computed.
3. Depending of the given maximum velocity the actual barrier dimensions (values passed to the computational kernel) are computed.

3.5.1.2 Computational methods

velocities

In maps velocities are presented as transport velocities (Q / H) and in time histories for barrier-data they are represented as the 'real' upstream and downstream velocities.

contraction coefficients

At a barrier important loss of energy takes place, apart from the loss by bottom friction. This extra energy loss is taken into account by adding an extra term to the equation of motion. This term has the same form as the bottom friction term, with a contraction or discharge coefficient that depends on the flow condition, such as subcritical, supercritical and gate-restricting flow. Furthermore, the energy loss can depend on the flow direction, as the sluice structure may not be symmetrical, so a contraction coefficient has to be given for each direction.

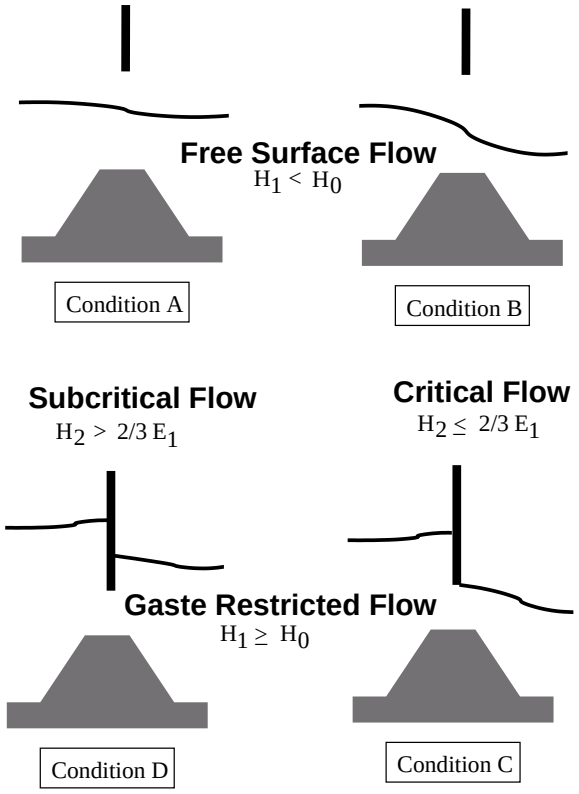


Figure 3.15: Different contraction coefficients may be specified for each of the four possible conditions for barrier flow

Hence, with 4 x 2 contraction coefficients all possible flow conditions can be described. The contraction coefficient for gate restricted subcritical flow may also depend on the orifice height ("hefhoogte") of the barrier. The contraction coefficients are in many cases known from model experiments or from field measurements by means of calibration with a Q-h relation (see below). The appropriate contraction coefficients are selected by the program from the computed flow condition, flow direction and possibly gate opening (meant for gate-restricting subcritical flow only).
In case of supercritical flow, the flow rate through the barrier ends up at a maximum value: the critical flow rate. Ongoing decrease of the downstream water level does not affect the flow through the barrier anymore. In WAQUA this "blocking" of the flow is modeled by substituting the equation of motion for the velocity point, at which the barrier is located, by the flow rate relation valid for the barrier (see below). In this manner, the advisable energy loss will be achieved in an alternative way. As a matter of fact, the same contraction coefficients are playing their part in this.

height, depth and width

The gate height, the sill depth and the relative width restriction (width/grid size) are all time-variable inputs. The gate height is described as positive upward with respect to the reference level, the sill depth as positive downward (by default). Thus if the sill is located at a depth of 5.0 m, then the barrier will be closed completely when the gate height is -5.0 m.

large- or small-scale modelling

The effect of the gate and the sill on the flow near a barrier can be modelled in two ways: large-scale ("overzichtsmodel") or small-scale ("detailmodel"). In case of large-scale modelling, the size of the barrier is assumed to be relatively small compared to the gridsize, so that the presence of the gate and the sill has no effect in the model. Only the energy loss across the barrier will be computed. For a TRIWAQ model, this means that all layers at the barrier are set open. As a consequence, the vertical velocity profile at the barrier remains unchanged. This modelling is equivalent to the formulation implemented in WAQUA. In case of small-scale modelling, all layers, except those between gate and sill, are set closed by means of setting the flow at the barrier to zero. This implies that the vertical structure of the flow near the barrier becomes more realistic. This modelling is appropriate for 3D applications in which the distribution of salt needs to be solved.

Note: It is recommended to decrease the depth at the barrier to the same level as the sill depth when the small-scale modelling is employed. This should not be done if the large-scale formulation is applied.

3.5.2 Barrier-barrier constructions

Some hydraulic constructions are more complicated than the barriers described in the previous section. Two examples are the culvert and sluice structures, shown in Figures 3.16 and 3.17.

Such structures are modeled as the combination of two barriers, which are located at the same grid point.

When only the bottom one of the barriers in a barrier-barrier structure is submerged, the construction is no different from a regular barrier and is treated as such by WAQUA.

When both barriers in such a barrier-barrier structure are submerged, WAQUA calculates the discharge through both barriers and adds these to obtain the discharge through the structure. WAQUA always enters this discharge into the model, even when the flow is subcritical. Calculating the barrier effect into an extra bottom friction term would be too complicated.

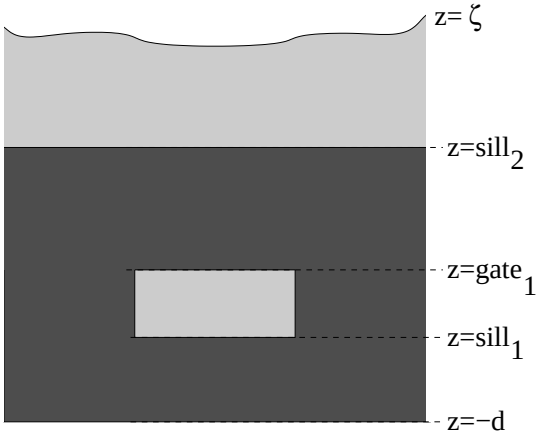


Figure 3.16: Frontal view of a culvert, which can be seen as the combination of a narrow barrier with a low sill and a low gate, on top of a wide barrier with a high sill and no gate.

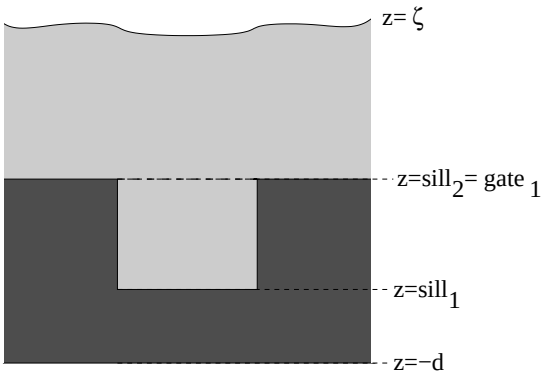


Figure 3.17: Frontal view of a sluice, which can be seen as the combination of a narrow barrier with a low sill and a low gate, on top of a wide barrier whose sill corresponds to the lower barrier's gate.

3.5.2.1 Weirs

3.5.2.2 Features

The modeller may define certain grid points to be weirs, which are fixed, non-movable constructions causing energy losses due to constriction of flow in the u-direction or the v-direction.

Weirs are commonly used in river simulation models. They are necessary to model sudden changes in depths, such as the border of a sand pit. Also groynes are modelled as weirs. A river simulation model can contain up to ten thousand weirs.

location

A weir is located at a velocity point and has to be specified as a U-flow or V-flow weir. An U-flow and a V-flow weir can have the same (m, n)-coordinates. Each of them works independently from each other. Each rectilinear weir is one grid space long. It is also possible to define diagonal weirs. A diagonal weir is a combination of an U- and V-weir. The length of a diagonal weir is the square root of the sum of the squared lengths of the U- and V-weir. A diagonal weir gives a better representation of the length of a skew weir than the combination of a single U- and a single V-weir.

monitoring

The user may request both so-called "maps" and "time-histories" for weirs. Maps concern a representation of the flow state at all weirs in a calculation, and are written to the output (SDS-) file at map-times. The variables stored here are the flow condition (sub- or supercritical), discharge, energyloss, and estimated flow-through height and velocity at the weir.

Time histories concern the same variables, plus the waterlevel and stream velocity on both sides of the weir, for a selection of the weirs. This allows to write these values (much) more frequently than the map-arrays.

Note: for diagonal weirs, the quantities in "diagonal" direction are stored instead of the components in grid direction (x-component for an U-weir and y-component for a V-weir).

3.5.2.3 Computational methods

The extra energy losses, caused by the weirs, are inserted directly into the momentum equation per grid cell. The calculation of this extra energy loss takes place outside the computational routines, and therefore it is not possible to use iteration steps in these routines. This can result in a less accurate solution and can lead to unstable behaviour. In case of a steady flow the user can use the

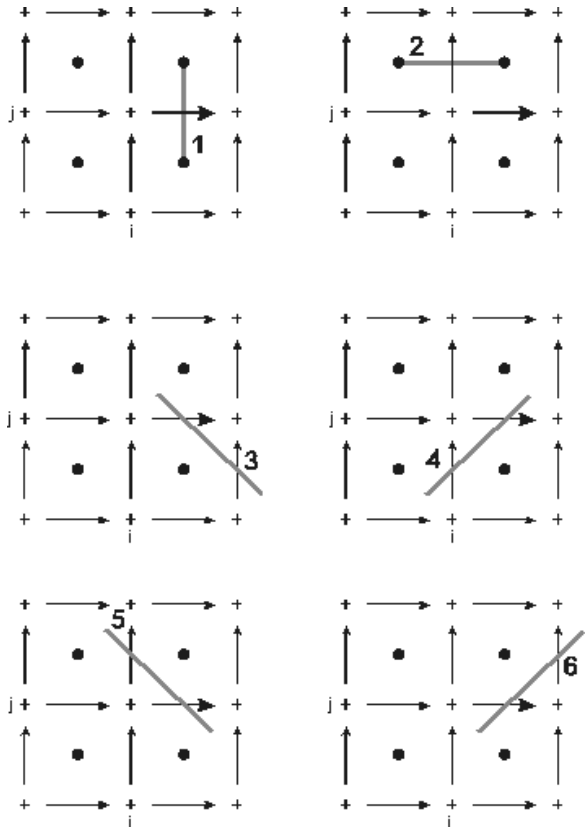


Figure 3.18: U-weir (left), V-weir (right).

parameter THETAC to reduce instable behaviour. For a description of this parameter see the user's guide pre-processor WAQPRE.

The extra energy loss in the direction rectilinear on the weir is known as ΔE. The calculation method for the energy loss depends on the flow condition (subcritical or critical flow) and on the mathematical model used (Wijbenga model or Villemonte model: see also section 2.9, Weirs, of WAQUA/TRIWAQ two- and threedimensional shallow water flow model, Technical documentation (SIMONA report 99-01).

3.5.2.4 Weirs in TRIWAQ

Since the introduction of vertical domain decomposition (DDVERT01) it is also possible to specify weirs in TRIWAQ with one layer. This is in particular of importance at TRIWAQ models with vertical refinement, which are partitioned into one or more sea domains (without weirs, in which in general KMAX > 1), combined to contiguous river domain(s) (containing weirs, in which KMAX = 1).

Since January 2008, it is also possible to specify weirs in TRIWAQ models with multiple layers. In this case the calculations for weirs as described above are performed using depth averaged quanti-

ties. The resulting energyloss is then put into the momentum equation per layer, and introduces an auxiliary term comparable to the waterlevel gradient.

3.6 Drying and flooding (tidal flats)

3.6.1 Features

The simulation accounts for tidal flats by considering grid points to be dry at depths less than a given marginal depth, DEPCRIT. When a velocity point becomes dry, it is taken out of the computation. When a water level point becomes dry the water level point and the four surrounding velocity points are taken out of the computation. For velocity points a check is made for flooding again at each (half) time step. Drying and flooding follows the sides of grid cells. It is a discontinuous movement of the closed boundary and may generate small oscillations in water levels and velocities; see Stelling and Wiersma, 1986.

3.6.2 Computational methods

Dynamic drying and flooding of points in the computational grid (i.e., activating and deactivating computational points during the simulation) is used to simulate tidal flats.

mask arrays	Mask arrays are used to determine the dry or wet status of a grid point. The mask arrays for the u- and v-velocity are KFU and KFV, respectively. A temporarily dry point is assigned a value of zero. If a point changes its status, e.g. from dry to wet, then also the mask array changes. Next to this, there are the index arrays KCU and KCV, indicating if a velocity point is part of the computational domain. A point which is permanently dry, is assigned a value of zero.
drying/flooding criteria	The status (wet or dry) of an active velocity point depends on the total water depth H. Because WAQUA is based on a staggered grid, see Fig. 3.19, the total water depth at velocity points must be determined from the surrounding depth and water level points. Usually the average depth is taken on the boundary of a grid cell, see Fig. 3.20. The water levels at each side of the boundary are averaged and the average depth along the boundary is added. In the neighbourhood of steep bottom gradients averaging the water levels may lead to an inaccurate determination of the total water depth. Drying occurs if for the total water level holds: $S + D < 0.5 \cdot$ DEPCRIT and wetting occurs if: $S + D > 0.5 \cdot$ DEPCRIT where S denotes the water elevation and D denotes the water depth. The location of the four surrounding depth points for a water level point is given in the following figure:

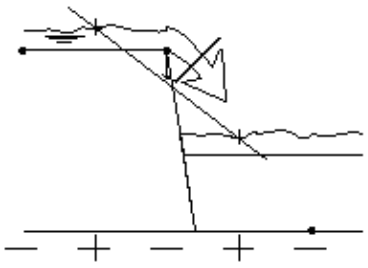


Figure 3.19: Drying of tidal flat - average approach

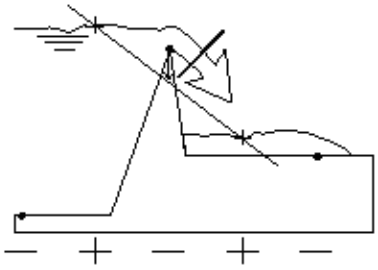


Figure 3.20: Overtopping of river bank - average approach

Fig. 3.19 shows drying of a tidal flat. When the tide falls and the water level in the channel falls below the depth of the tidal flat, averaging the water levels at the velocity points leads to drying. A large volume of water is left on the tidal flat. The storage capacity is overestimated. Fig. 3.20 shows the situation of a river which overtops its bank. In case of the average approach the total water depth at the bank is negative and the velocity point remains dry. The water level will rise too much in the main channel of the river leading to unrealistic water levels downstream. Ultimately the average water level fulfills the flooding check, the flood plains are suddenly filled with water, leading to a large disturbance in the flow field.

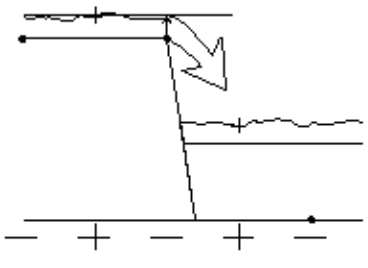


Figure 3.21: Drying of tidal flat - upwind approach

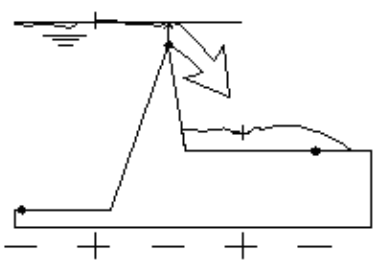


Figure 3.22: Overtopping of river bank - upwind approach

For this kind of simulations Stelling (1984) and Van der Molen (1997) introduced an 'upwind' approach to determine the total water depth at velocity points and a 'tiled depth' approach for depths that are specified at water level points. The upwind water level is used for velocity points, which are wet. In dry cell boundaries, the maximum of the two water levels is taken. Fig. 3.21 and Fig. 3.22 give an impression of this approach. The 'upwind approach' is switched on via keyword UPWIND_ZETA. In the past, the 'upwind' approach was switched on if the total depth, which was determined by the 'average' approach, became less than DUPWND. The difference is that with the flag UPWIND_ZETA in all grid points upwinding is either applied or not applied, while with the DUPWND parameter only in the grid points with a sufficiently low water level this was applied. The 'tiled depth' approach is switched on via keyword METH_DPUV (see section 3.6.2).

In WAQUA, the modeller may define certain velocity points to be weirs. Weirs are constructions causing energy losses due to constriction of the flow. The height of the edge of the weir, the crest, is taken into account in the drying and flooding control. An upwind approach is used to determine the difference between the water level and the crest of the weir. The 'upwind' water level is used for velocity points, which are wet. At a dry weir point, the maximum of the two neighbouring water levels is taken.

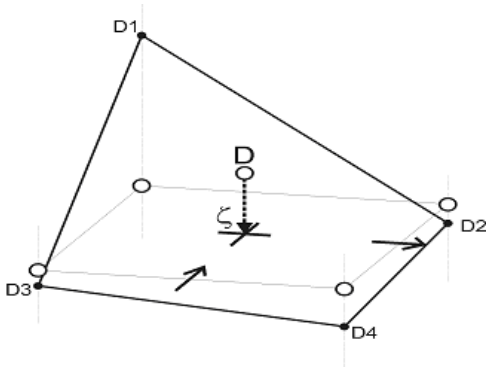


Figure 3.23: Depth in water level point - mean criterion

One or more positive total water depths at the grid cell boundaries do not guarantee a positive total water depth in the centre of the cell. This is illustrated in Fig. 3.23, where in the centre of the cell an average depth is taken. Positive control volumes are necessary for transport simulations. Therefore drying is not only checked at the cell boundaries, but also at the water level location at the centre of the cell. The depths D1, D2, D3 and D4, however, are taken positive in downward direction of the reference level, which means that H = ζ + D is the total local depth. The water level ζ is taken positive above the reference water level.

depth in water level points

The computed water depth D in a water level point depends on the four neighbouring depth points (see) and on keyword METH_DPS:

METH_DPS = 'MAX_DPD': maximum criterion, with

$$D = max\,(D1, D2, D3, D4)$$

METH_DPS = 'MAX_DPUV': maximum criterion, with

$$D = max\left\{\frac{(D1+D2)}{2}, \frac{(D2+D4)}{2}, \frac{(D4+D3)}{2}, \frac{(D3+D1)}{2}\right\}$$

METH_DPS = 'MEAN_DPD': average criterion, with

$$D = \frac{D1+D2+D3+D4}{4.}$$

METH_DPS = 'MIN_DPUV': minimum criterion, met

$$D = \min\left\{\frac{(D1+D2)}{2}, \frac{(D2+D4)}{2}, \frac{(D4+D3)}{2}, \frac{(D3+D1)}{2}\right\}$$

If the list is followed from the top to bottom, than the bottom becomes more shallow.

depth in velocity points

The computed water depth D in a velocity point depends on keyword METH_DPUV:
METH_DPUV = 'MIN_DPS', with

$$D = \min\left(D^S_{m,n}, D^S_{m+1,n}\right)$$

METH_DPUV = 'MEAN_DPS', with

$$D = \frac{D^S_{m,n} + D^S_{m+1,n}}{2}$$

METH_DPUV = 'MAX_DPS', with

$$D = \max\left(D^S_{m,n}, D^S_{m+1,n}\right)$$

METH_DPUV = 'MEAN_DPD', with

$$D = \frac{D^d_{m,n} + D^d_{m+1,n}}{2}$$

in which the superscript indicates whether a water level point (s) or a depth point (d) is involved.

drying in water level points

A grid cell is set dry if one of the surrounding velocity points is still wet and if for the total water depth in the centre of a computational cell holds that:

$$\zeta + D < 0.5 \cdot THRES_WL$$

When drying occurs for the extra drying check in water level points, then in the four surrounding velocity points the velocity is set to zero.

In TRIWAQ a velocity point can only be wet if for the two surrounding water level points holds that:

$$\max(\zeta_m, \zeta_{m+1}) > -\min(D_m, D_{m+1}) + THRES_WL$$

which means that the water level in the wet point should be above the bottom in the dry point.

drying in velocity points

At the end of each halve time step in the ADI time integration method the new total water depth is computed on the basis of the velocity points, in which the water levels follow the ADI method (see Section 3.4.2.1).
A wet velocity point becomes dry if the total water depth H < 0.5 THRES_UV, in which THRES_UV is the threshold value. At the start of each halve time step in the ADI time integration method a dry velocity point becomes wet, in the direction in which the velocity component is treated implicitly in time (according to the ADI method), if the total water depth H > THRES_UV. The difference of a halve threshold value between flooding and drying prevents to a large extent that drying and flooding occurs in consecutive time steps (the so-called "flip flop" effect). The velocity at a dry cell interface is zero. A complete grid cell is taken out of the computation if the four surrounding grid cell interfaces have become dry.

For more details about the drying and flooding control of WAQUA and TRIWAQ, see Van Kester and De Goede (1997).

compatibility

In previous WAQUA/TRIWAQ versions the keyword METH_DPS was not used, but keyword IDRYFLAG. Keyword IDRYFLAG is forbidden in the current version. Below we have specified how the old keyword IDRYFLAG can be replaced by values of the new keyword METH_DPS:

IDRYFLAG = 0: METH_DPS = 'MAX_DPUV'
IDRYFLAG = 1: METH_DPS = 'MEAN_DPD'
IDRYFLAG = 2: METH_DPS = 'MIN_DPUV'
IDRYFLAG = 3: CHECK_WL = 'NO', METH_DPS = 'MEAN_DPD'

Note: In case of constituents the latter two options (IDRYFLAG=2 and IDRYFLAG=3) are not allowed.

For compatibility with the option IDRYFLAG=3 the keyword CHECK_WL has been introduced, in which two options are available:

CHECK_WL = 'NO' : no extra drying check in water level points
CHECK_WL = 'YES' extra drying check in water level points

References:

1. Stelling, G.S., 1984. On the construction of computational methods for shallow water flow problems. Rijkswaterstaat communications, No. 35, The Hague, Rijkswaterstaat.
2. Stelling, G.S., A.K. Wiersma, and J.B.T.M. Willemse, 1986. Practical aspects of accurate tidal computations. J. Hydr. Eng. Div., ASCE.
3. Van der Molen, J., 1997. Tides in a Salt-Marsh. Proefschrift Vrije Universiteit Amsterdam.
4. Van Kester, J.A.Th.M. en E.D. de Goede, 1997. Verbeteren van algoritme voor droogvallen en onderlopen in WAQUA en TRIWAQ. WL-rapport Z2292.
5. Technisch Rapport TR04-02, 2004. Detailontwerp verbetering droogvallen en onderlopen in WAQUA / TRIWAQ dr. ir. E.A.H. Vollebregt, dr. ir. B. van 't Hof (VORtech), ir. J.A.Th.M. van Kester (WL | Delft Hydraulics)
6. Memo EV/M04.100, 2004. Overzicht van nieuwe droogvalopties voor gebruikers van WAQUA / TRIWAQ Erik de Goede (WL | Delft Hydraulics), Edwin Vollebregt en Bas van 't Hof (VORtech Computing)

3.7 Harmonic analysis of tides

3.7.1 Features

tidal motion

The modeller may want to analyse tidal motion in a shallow coastal sea area where the data are obtained from a numerical simulation by means of WAQUA or TRIWAQ. The purpose of tidal analysis is to make sense of a time series of tidal elevation (or current) observations by mapping them into the amplitude and phase of a number of tidal components. The obtained results, i.e. amplitudes and phases throughout the grid, fully determine the astronomical tide and are written to the SDS-file. They can be used for predicting the water level or current for any period.

3.7.2 Computational methods

In this section an overview is given of the tidal analysis of computed water levels within a model obtained with WAQUA or TRIWAQ.

3.7.2.1 Mathematical representation of the tide

astronomical tide

The astronomical tide can be described in terms of a series of simple harmonic components, each with its own frequency ω (rad/min). This representation is given by

$$\zeta(x,t) = A_0(x) + \sum_{i=1}^{K} A_i(x) f_i \cos\left(\omega_i t + (V_0+u)_i - g_i(x)\right)$$

where,

$\zeta(x,t)$	=	tidal water level at location x and at time t
$A_0(x)$	=	mean water level at location x
$A_i(x)$	=	astronomical amplitude at location x
f_i	=	nodal amplitude factor
ω_i	=	angular velocity
$(V_0+u)_i$	=	astronomical argument
$g_i(x)$	=	improved kappa-number or local phase lag at x

WAQUA offers the user the possibility to approximate the tidal water current with the above formula.

harmonic components

The angular velocities are determined by the movement of the sun-earth-moon system and are considered known. There are 195 commonly used harmonic components available; see Appendix C. From a purely analytical point of view the above formula could be used to determine any number of components from a very short time series. However, certain basic rules exist. In general, the longer the period of tidal data to be included in the analysis, the larger the number of components which may be independently determined. Hence, if the number of components is given, then a minimum acceptable length of time series is required. It is common practise to rely this on the Rayleigh criterion which states that the minimum length of a time series required to isolate two components apart in frequency $\Delta\omega$ is $T_{rc} = 2\pi/\Delta\omega$. For example, with components M2 (.008431 rad/min) and S2 (.008727 rad/min), we have T_{rc} = 14.7 days. Hence, if the modeller chooses a certain number of harmonic components for a given simulation period (= T_{sim}), the program will check whether the criterion is violated or not. If it is violated, i.e. $T_{sim} < T_{rc}$, a warning message will follow.

astronomical splitting

Where the length of time series is too short to separate two important components, it is usual to relate the amplitude and phase lag of the weaker component to the amplitude and phase of the stronger one. These relationships are called the splitting factors and are given by

$$r_A = \frac{A_{related}}{A_{reference}}, r_g = g_{related} - g_{reference}$$

These splitting factors may only be used in case of the equilibrium tide. For example, to separate S2 from the weaker K2 requires 182.6 days. This is too long for most WAQUA applications. Instead of including the components S2 and K2 in the analysis, we applied the astronomical splitting. This means that only the reference component S2 is incorporated in the analysis. The component K2 is coupled to that component and solved implicitly as part of S2. Afterwards, the two components S2 and K2 are decomposed using the splitting factors r_A = 0.2723 and r_g = 0. Table 3.1 gives a set of four related components and the (equilibrium) splitting factors, which is effective for the analysis of water levels along the Dutch coast.

Component	related to	r_A	r_g
P1	K1	0.33082	0.0
NU2	N2	0.19386	0.0
K2	S2	0.27230	0.0
LABDA2	2MN2	0.26295	0.0

Table 3.1: A set of related components with splitting factors

These default values can be used in WAQUA. However, WAQUA allows the user to employ the space-varying splitting factors.

nodal modulations

The quantities f_i and u_i are nodal modulations and vary from year to year. These modulations can be found from tide tables or can be computed empirically; see, for example, P. Schureman, *Manual of harmonic analysis and prediction of tides*, U.S. Coast and Geodetic Survey (1941). Furthermore, V_0 is the phase of the component in the equilibrium tide at t = 0. It should be noted that within WAQUA both nodal amplitude factor and astronomical argument at a given year of the time series with respect to January 1st, 1900, 0000 h are calculated.

tidal constants

The space-varying quantities A_0, A_i and $0 \leq g_i \leq 2\pi$ are called tidal constants. They represent the character of the tide. The tidal constants are written to the SDS-file.

3.7.2.2 Harmonic analysis

If for a specific location the tidal constants A_0, A_i and g_i are known, the above formula can be used to predict the local water level ζ at any time. Conversely, if at a location a set of $n_l - n_f + 1$

observations $\{h(t_n)|n = n_f, ..., n_l\}$ is known, the above formula can be employed to calculate the tidal constants as good as possible. In case of K relevant harmonic components, a total of 2K+1 unknowns A_0, A_i and g_i must be determined. This is realised by minimisation of the variance

$$\sum_{n=n_f}^{n_l} \left(h(t_n) - \zeta(t_n)\right)^2$$

least squares method

by means of the least squares method. This gives a linear set of 2K+1 equations in the unknown variables. By building the system of equations in an efficient manner, an equidistant time series is assumed, thus $t_n = n\Delta t$. Commonly used time steps are 10, 15, 30 or 60 minutes. Furthermore, in order to have a non-singular solution, the time step is restricted to

$$\Delta t \leq \frac{\pi}{\omega_{\max}}, \omega_{\max} = \max\left\{\omega_i \,|i = 1, \ldots, K\right\}$$

The system of equations is solved by means of a Choleski decomposition.

3.8 Transport of constituents

3.8.1 Features

constituents	Constituents are such various properties as salinity, chlorosity, temperature, dye and other dissolved substances. The water quality feature of the simulation accepts time-varying concentrations of constituents at open boundaries and at sources, and computes transports and concentrations throughout the field. In future constituents which are interactive will be introduced. Specific identification is made of salinity (and temperature).
interpolation	Like all time-varying inputs, constituent concentrations are interpolated across time, and input concentrations at open boundaries are also interpolated across space. Both in space and in time, linear interpolation is used.
time interpolation	Interpolation across time occurs for the time-varying input of concentrations, which can be equidistant or not. The value at the begin-ning of interpolation is either the previous value given in the input, or the initial value (given in section TRANSPORT, subsections BOUNDARIES and DISCHARGES in the WAQPRE user's guide).
space interpolation	Interpolation across space occurs at open boundaries, where inputs are given at the two ends of the opening, and values at intervening grid points on the opening are calculated by interpolation. Space interpolation is done to constituent concentration and - only initially - constituent return time at open boundaries.
user routines	The user can influence the transport computation with the feature of user defined 'user routines' (see section 3.9). In this way interaction of constituents for example can be introduced.
equation of state	When salinity is defined as a special constituent, densities can be computed with help of the equation of state (see section 3.4.2.7). These densities can be used in the flow computation in baroclinic mode.

3.8.2 Computational methods

In this section an overview is given of the time advancement procedure of state variables and accuracy considerations. Furthermore, forcing functions that drive the simulation are mentioned.

For the computation of concentrations of the various constituents the grid of the flow computation is used. Values of constituent concentrations, boundary conditions and sources and sinks are located in the water level points (see section 3.2). In case of constituents only option 1 and 2 in the drying and flooding procedures is allowed (see section FLOW, subsection DRYING in the WAQPRE users's guide). Options 0 and 3 will internally be overruled by option 1.

3.8.2.1 Time advancement of state variables

Water levels and u- and v-velocity components, resulting from the flow computation are used in the constituent transport computation together with the depths and discharge rates in local and global sources and sinks. For a rectilinear grid the following transport equation is used (see also section 3.3.2.2 and section 3.3.3.2 for other grids):

$$\frac{\partial}{\partial t}(Hc_l) + \frac{\partial}{\partial x}(Huc_l) + \frac{\partial}{\partial y}(Hvc_l) =$$
$$\frac{\partial}{\partial x}\left(HD_x\frac{\partial c_l}{\partial x}\right) + \frac{\partial}{\partial y}\left(HD_y\frac{\partial c_l}{\partial y}\right) + S^+c_l^+ - S^-c_l + SC_l^{\exp} + S^{imp}c_l$$

with:

u, v	=	velocity components in x- en y-direction in m/s;
c_l	=	concentration of constituent number l;
S^+	=	discharge in local source in m^3 / s;
c^+	=	constituent concentration in local source;
S^-	=	negative discharge in local source (= sink) in m^3 / s;
S^{imp}	=	implicit global source/sink per unit area in (m^3 / s) / m^2;
SC^{exp}	=	explicit global source/sink per unit area in (m^3 / $s \cdot$ [C]) / m^2;
D_x, D_y	=	diffusion coefficient in u- and v-velocity points in m^2 / s;
$H = h + \zeta$	=	total water depth in m.

state variables

The integration proceeds in increments of half time steps (half of TSTEP), in sequence with the flow computation. The state variables are the constituent concentrations. They are computed in each half time step in water level points of the staggered grid, starting from user given (space varying) initial concentration fields.

discretisation

The first term in the transport equation is the rate of change of the mass, located in water level points. Advective and diffusive mass fluxes are computed in u- and v-velocity points, using interpolated concentrations. The second and third term represent (horizontal) advection, the first and second term on the right hand side (horizontal) diffusion. The remaining terms are local and global sinks and sources, located in water level points. In this way a control volume approach is used in a water level point (see also section 3.2.1.2).

ADI scheme The numerical method is an ADI scheme. The method has as properties:

- it is symmetrical over the x and y directions;
- it attains its highest accuracy over every two half time steps;
- it is mass conserving;
- it is computationally efficient;
- it is suitable not only for time-dependent problems, but also for steady state problems;
- it is unconditionally stable.

accuracy

This last property, however, does not imply that the time step can be chosen without bounds. The quality of the solution is also dependent on the boundary conditions, the forcing functions and other time-dependent input. Therefore, in practice the accuracy puts a limit on the time step (TSTEP). In many problems the Cell Peclet number Pe and the velocity based Courant number C_u are relevant measures for accuracy. For equal spatial steps (dx = dy) they can be defined by:

$$Pe_\Delta = \frac{u\Delta x}{D} \quad \text{and} \quad C_u = \frac{u\Delta t}{\Delta x}$$

where:

u	=	flow velocity in m / s;
D	=	diffusion coefficient in m^2 / s;
Δx	=	spatial step (grid space) in m;
Δt	=	time step in s.

The transport equation is solved on a numerical grid using central differences. This means that artificial/numerical diffusion is absent, but also that unwanted oscillations and negative concentrations can appear when diffusion/dispersion is low and strong concentration gradients are present in the model, especially when $Pe_\Delta > 2$. Strong gradients can appear near sources and near open boundaries with strong varying boundary conditions at inflow.

The Courant number C_u determines the accuracy of the propagation of constituents. This does not result in an extra constraint concerning the time step used in the flow computation.

For an accurate transport computation, also the results from the flow computation must be mass conservative. This is determined by the continuity equation in the flow equations, which is solved iteratively using the parameters ITERACCURVEL/WL and ITERCON (see subsection METHODVARIABLES of section FLOW in the WAQPRE user's guide). These parameters must be chosen carefully (e.g. 10^{-6} and 20 respectively).

There is no time smoothing present in the integration procedure.

3.8.2.2 Forcing functions

The main forcing function for transport of constituents is the computed flow field (see section 3.3).

Constituent mass can enter the model at the location of the sources, depending on user defined discharge rates and concentrations. At sinks (sources with negative discharge rates) constituent mass is taken out of the model, depending on user defined discharge rates and computed concentrations

at the location of the sink.

With the feature of user defined 'user routines' (see section 3.9) the user can define global sources and sinks to be used in the transport computation.

Boundary conditions are necessary at open boundaries, dependent on the direction of current flow relative to each point on the opening. At outflow concentrations are determined from the computed inner concentrations. At inflow the (time-varying) background concentration is prescribed. A smooth (sinusoidal) transition from the outflow concentration to the user prescribed inflow boundary condition is possible. Diagonal open boundaries are treated as a horizontal as well as a vertical open boundary.

The direction of the flow is determined as follows:

Boundary location	Criteria	Flow direction
Left	u positive	Inward
Left	u negative	Outward
Right	u positive	Outward
Right	u negative	Inward
Top	v positive	Outward
Top	v negative	Inward
Bottom	v positive	Inward
Bottom	v negative	Outward

The concentration in each open boundary point is computed as follows:

- outward flow: boundary concentration is determined by pure advection of inner concentration:
  $\frac{\partial c_1}{\partial t} + u\frac{\partial c_1}{\partial x} = 0$ or $\frac{\partial c_1}{\partial t} + v\frac{\partial c_1}{\partial y} = 0$
- inward flow: boundary concentration is computed by (Thatcher-Harleman):
  $c_1 = c_1^{out} + \left(c_1^{bnd} - c_1^{out}\right) \frac{1}{2}\left\{\cos\left(\pi\frac{\Delta T}{T_{ret}}\right) + 1\right\}$
  with:
  c_l = concentration of constituent number l;
  u, v = velocity component at outflow boundary in m/s;
  c_l^{out} = computed concentration at open boundary at the time that flow turns inward;
  C_l^{bnd} = ambient (or background) concentrations at opening;
  ΔT = elapsed time since flow turned inward;
  T_{ret} = duration of the return cycle (constituent return time).

The return cycle duration is set equal to the maximum time in the cycle (ΔTret) when the flow is outward. Therefore, each time that inward flow is interrupted, even briefly, the return cycle duration is reset to $\Delta\text{Tret} \cdot C_l^{out}$ is set equal to the concentration at the point when the flow turns inward. The concentration at the point will return to C_l^{bnd} during the interval ΔTret if inward flow is uninterrupted. This is illustrated in Fig. 3.24

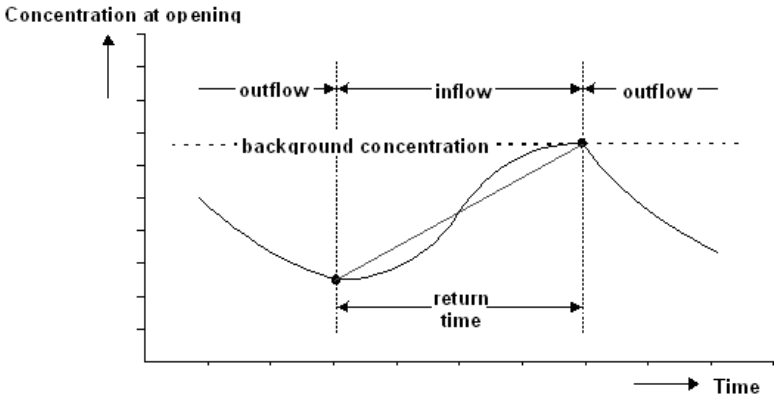


Figure 3.24: Constituent return cycle

3.9 User routines

3.9.1 Features

An increasing percentage of hydraulic modelling activities is carried out to learn more about the behaviour of in water dissolved substances and on the reactions between these substances.

The range of possible applications in this field is enormous. Examples are water quality studies, modelling flocculation processes, turbulence, resuspension, sedimentation, temperature, spreading of algae etc.

The user routine is a feature within WAQUA that provides the inves-tigator with the possibility to use the by WAQUA computed water movement in combination with his self defined water quality model.

Possibilities:

- interactions between constituents;
- interactions between constituents and properties fixed to the bottom;
- reactions between bottom related parameters;
- adjustable diffusion coefficients.

The reaction kinetics as well as the involved parameters and boundary conditions are under the user's control.

3.9.2 Mathematical description

The following equations govern the transport computation. The user routine gives access to the diffusion coefficient, the global source and the global sink.

$$\begin{gathered}\frac{\partial}{\partial t}(hc)+\frac{\partial}{\partial x}(huc)+\frac{\partial}{\partial y}(hvc)+\frac{\partial}{\partial z}\left(h(w-w_{val})c\right)\\ =\frac{\partial}{\partial x}\left(D_x\frac{\partial}{\partial x}(hc)\right)+\frac{\partial}{\partial y}\left(D_y\frac{\partial}{\partial y}(hc)\right)+\frac{\partial}{\partial z}\left(D_z\frac{\partial}{\partial z}(hc)\right)\\ +S_+-Lc\end{gathered}$$

where

c	=	constituent concentration in [c]
D_x, D_y	=	horizontal diffusion coefficient in m^2/s
D_z	=	vertical diffusion coefficient in m^2/s
S	=	global source (explicit) per unit area in [c] $\cdot\, m^3/s/m^2$
L	=	global sink (implicit) per unit area in $m^3/s/m^2$

3.9.3 Data input/output

To model water quality processes input data is necessary. Apart from the water movement which is computed by WAQUA the user should be enabled to feed the model with parameters he wants to take into account.

In WAQPRE the user can control the following flexible data blocks:

- real or integer data stating for instance a reaction constant;
- time series for instance describing the inflow of energy due to a fluctuating radiation of the sun;
- spatial distributed data like the initial thickness of a layer of sediment on the bottom of the model;
- time and space dependent data which can be used to describe the seasonal variation in the growth rate of local bottom vegetation.

3-D data like the magnitude of a 3-D space dependent coefficient can be offered to the user routine in layers using the spatial distributed data block.

For detailed information refer to the user's guide preprocessor WAQPRE, USERDATA_TRANSPORT.

WAQPRE will store the user defined data and create work space within the data structure for fields of output data, defined by the user, but computed during simulation.

In this way the SDS file will, after computation, contain results of user defined parameters.

Within the user routine the unit numbers of the WAQPRO report - and message file are known. Using this information warnings and errors occurring during execution of WASUST can be taken care of in a structured way.

3.9.4 Computation

By a linking procedure the user routine programmed and compiled by the user becomes part of the WAQUA WAQPRO executable. The name of the routine to be made by the user is WASUST. The heading of this routine is fixed, refer to the chapter on WASUST in the user's guide processor WAQPRO.

Within this user routine a part of the local WAQUA data structure is made available to the user. A list of the available data within WASUST is given in the user's guide processor WAQPRO. It contains time dependent data computed by WAQUA as well as a number of fixed data entries, refer to common CWAUST.

Within WASUST the SIMONA processing tool SIFODE can be used to enable the user to compute the interaction between dissolved sub-stances and parameters fixed to a given location. In general, SIFODE controls the change in the amount of bottom fixed substances.

An example in which case SIFODE should be used is the interaction between silt, which is modelled as a dissolved constituent, and the thickness of a layer of silt in a given grid cell.

WASUST is called by the WAQUA system every half time step, just before the advection diffusion solver.

Programming the content of WASUST is in fact only limited to the creativeness of the user.

3.9.5 Generating procedure

Within WAQUA the procedure `gen_waqpro.pl` is available that:

- compiles the WASUST routine;
- links the result to the object modules of WAQPRO;
- makes a WAQPRO executable containing the user routine.

For more information on how to use the WAQPRO generating proce-dure refer to the chapter on GEN.WAQPRO in the quick reference user's guide.

3.10 Data flow

3.10.1 Input

WAQPRE input file	Input for a simulation is given to the pre-processor WAQPRE by means of the WAQPRE input file. The modelling features that are described in previous sections can be defined or selected in this file. Other inputs which can be given are mostly control parameters that select certain data or request certain displays. Actually, some of the simulation input data are also control parameters choosing various options in plotting maps, and selecting the simulation time intervals at which to compute and print and plot.
restart	A simulation can also be 'restarted ', which means that the results of a previous WAQUA experiment on the SDS file are used as input to the WAQUA processor WAQPRO. The initial data for WAQPRO will be exactly the same as the data used in the simulation at the time level specified for restart, without any loss of information.
WAQWND file	In case of a computation using space varying wind and pressure a wind SDS file is needed. It is created by WAQWND using a KNMI wind file.
program user's guides	Descriptions here of inputs to individual programs are only indicative of data preparation entailed. The reader is also referred to detailed, complete and physically ordered descriptions in individual program user's guides.

3.10.2 Output

prints and plots	The output from the WAQUA system consists of plots (both maps and time histories) and printed messages and reports. Plots are particularly valuable because they present a wealth of information in a highly visible form. Plots and reports generally are identified by program run ID's and run dates and times, for unambiguous cross-reference.
examples	Sample plots and prints can be obtained by running the available test models.
error and warning messages	All programs print run, error, and warning messages, as applicable. Run messages generally offer information when a program is running properly, while error and warning messages warn of problems found. Therefore, the run messages generally are similar from run to run of the same program, but the error and warning messages will differ greatly, if they appear at all.

3.11 CPU and memory usage for computational models

It is sometimes difficult to determine the amount of memory and cpu time that is needed for a certain simulation model. The amount of required cputime, wallclock time and memory usage for a number of specific simulations is given in Table 3.3. Also the relevant parameters of the simulations are included in Table 3.3. The parameters and properties of the used simulation models are given in Table 3.4. The calculations are done on several computers. The specifications of the processors are presented in table 3.2. One simulation is performed on a Xeon2 processor. For this system a specification is not available. All simulations are performed with SIMONA major release 2006-01.

A relation between required cputime and wallclock time and the number of grid points (mnmaxk) is given in figure 3.25. The cputimes and wallclock times are divided by the number of timesteps and the number of layers. The graphs are based on the values of the specific simulations of table 3.3, however they can be used as an indication for the required computing time. The values of the kalman simulation with the "kustzuid" model are not included in the graphs, as they don't fit inside the used scale.

Machine	Name	Cpu (MHz)	Mem (Mb)
A	AMD Opteron(tm) Processor 248	2191	2056
B	Dual Core AMD Opteron(tm) Processor 280	2391	8110
C	Intel®Pentium®4 CPU 2.26 GHz	2260	1284

Table 3.2 Processor specifications.

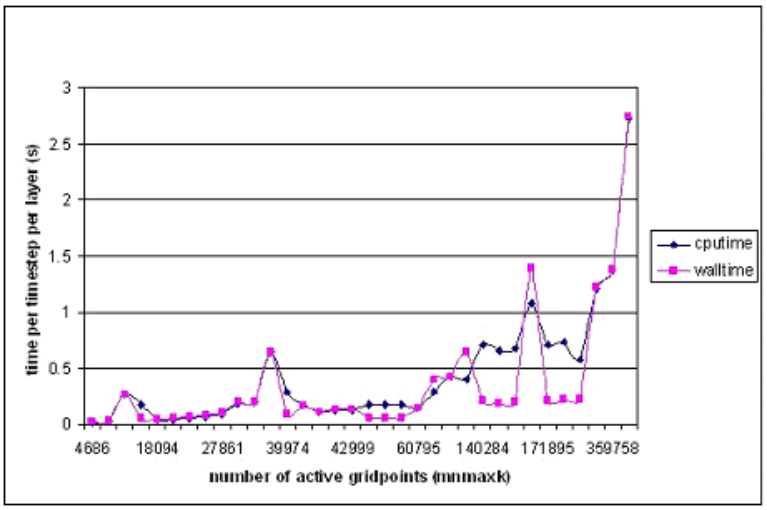


Figure 3.25: Cputime and wallclock time as function of mnmaxk (one timestep and one layer)

Model	Type of simulation	Computa-tional model	npart	Timestep	Nr of steps	mmax	nmax	mnmaxk	Cpu (s)	Wallclock (s)	Mem (mw)	Machine
dcsm-1998-v5	Astro	Waqua	1	10	576	201	173	19810	28	31	3.4	B
dcsm-1998-v5	Cold Start	Waqua	1	10	360	201	173	19810	22	24	6.1	B
dcsm-1998-v5	Restart	Waqua	1	10	274	201	173	19810	18	20	6.1	B
grevelingen-fijn-exvd-1991-v1		Waqua	1	1	4340	111	338	18094	212	213	3.4	A
grevelingen-grof-exvd-1991-v1		Waqua	1	1	4340	56	170	4686	47	100	1	A
IJmond -1999-v3	Astro	Waqua	4	0.5	11520	350	193	47101	2055	590	9.6	B
IJmond -1999-v3	Cold Start	Waqua	4	0.5	7000	350	193	47101	1265	352	9.5	B
IJmond -1999-v3	Restart	Waqua	4	0.5	5480	350	193	47101	997	279	9.5	B
kuststrook-fijn-1999-v4	Astro	Waqua	4	1	5760	942	402	140284	4082	1197	39.6	B
kuststrook-fijn-1999-v4	Cold Start	Waqua	4	1	3500	942	402	140284	2307	679	40	B
kuststrook-fijn-1999-v4	Restart	Waqua	4	1	2740	942	402	140284	1869	540	39.8	B
kuststrook-grof-1999-v4	Astro	Waqua	1	2	2880	488	147	27861	269	273	6.2	B
kuststrook-grof-1999-v4	Cold Start	Waqua	1	2	1800	488	147	27861	345	359	8.9	B
kuststrook-grof-1999-v4	Restart	Waqua	1	2	1370	488	147	27861	267	279	8.8	B
kustzuid-2004-v3	Kalman, 1 layer	Triwaq	1	1	4850	213	300	39974	136840	136875	25.8	C
kustzuid-2004-v3	Usegain, 1 layer	Triwaq	1	1	8640	213	300	39974	5534	5549	11.9	C
kustzuid-2004-v3		Waqua	4	1	8160	213	300	39974	2390	748	8.8	C
kustzuid - 2004-v4		Waqua	1	0.125	24960	1950	400	143844	26850	34766	60.2	A
lauwers-meer-2005-v1		Waqua	1	0.5	8640	230	557	81633	3657	3669	12.7	A
maas-hr2006_4-v1		Waqua	1	0.25	69120	187	6003	359758	95085	95120	78.2	Xeon2
markermeer-2005-v2		Waqua	1	0.1	13200	430	614	60795	1889	1917	26.1	A
ndb-2004-v1		Waqua	1	0.5	7200	501	1689	185342	8687	8837	69.5	A
nzk-2003-v1	4 layers	Triwaq	1	0.5	8640	124	566	16443	9330	9337	25.6	A
nzk-2003-v1	4 layers -salt	Triwaq	4	0.5	8640	124	566	16443	6197	1784	22	B
rijn-j95_4-v1		Waqua	1	0.25	57600	672	4273	640716	157199	158244	195.3	A
scaloost-fijn-exvd-v1		Waqua	1	0.25	17240	250	579	75255	4965	6943	12.8	A
scalwest fijn-v1		Waqua	1	0.5	8620	542	569	82387	3479	5579	25	A
zeedelta-2000-v8	Astro	Waqua	4	0.5	11520	501	1539	171895	8243	2471	67.9	B
zeedelta-2000-v8	Cold Start	Waqua	4	0.5	7000	501	1539	171895	5142	1527	67.8	B
zeedelta-2000-v8	Restart	Waqua	4	0.5	5480	501	1539	171895	3171	1193	67.7	B
zoommeer-fijn-1998-v1		Waqua	1	0.5	11260	195	642	40986	1843	1854	10.7	A
zoommeer-grof-1998-v1		Waqua	1	1	5630	98	322	10605	194	195	2.9	A
zuno-1999-v3	Astro	Waqua	1	2.5	2304	486	170	42999	257	261	8	B
zuno-1999-v3	Cold Start	Waqua	1	2.5	1440	486	170	42999	180	190	9.8	B
zuno-1999-v3	Restart	Waqua	1	2.5	1096	486	170	42999	137	145	9.7	B

Table 3.3 Time and memory requirements for a number of specific simulations

	dcsm-1998-v5	grevelingen-fijn-exvd-1991-v1	grevelingen-grof-exvd-1991-v1	ijmond-1999-v3	kuststrook-fijn-1999-v4	kuststrook-grof-1999-v4	kustzuid-2004-v3	kustzuid-2004-v4	lauwersmeer-2005-v1	maas-hr2006_4-v1	markermeer-2005-v2	ndb-2004-v1	nzk-2003-v1	rijn-j95_4-v1	scaloost-fijn-exvd-v1	scalwest-fijn-v1	zeedelta-2000-v8	zoommeer-fijn-1998-v1	zoommeer-grof-1998-v1	zuno-1999-v3
NDIMEN : dimension of problem (= 2 for WAQUA, = 3 for TRIWAQ)	2	2	2	2	2	2	3	2	2	2	2	2	3	2	2	2	2	2	2	2
MMAX : number of m grid points	201	111	56	350	942	488	213	1950	230	187	430	501	124	672	250	542	501	195	98	486
NMAX : number of n grid points	173	338	170	193	402	147	300	400	557	6003	614	1689	566	4273	579	569	1539	642	322	170
MNMAX : dimension : max of (NMAX,MMAX)	201	338	170	350	942	488	300	1950	557	6003	614	1689	566	4273	579	569	1539	642	322	486
MNMAXK : dimension : 1 + number of computational grid points	19810	18094	4686	47101	140284	27861	39974	143844	81633	359758	60795	185342	16443	640716	75255	82387	171895	40986	10605	42999
NENCLO : number of points in computational grid enclosure	601	377	195	192	3092	678	1108	3482	519	6923	950	4605	991	15301	1216	1159	4405	822	458	925
LDAM : number of dam points(permanent dry points)	230	0	0	46	137	21	49	49	64	0	0	0	151	0	216	0	239	65	16	67
NOCOLS : number of computational grid columns	450	187	93	390	2276	788	674	5586	470	2363	628	2693	1068	5169	728	1339	2602	577	283	862
NOROCO : number of computational grid rows and columns	869	436	218	597	4892	1328	1687	7238	1091	9024	1374	7972	2144	14506	1500	3131	7297	1393	687	1494
NOROWS : number of computational grid rows	419	249	125	207	2616	540	1013	1652	621	6661	746	5279	1076	9337	772	1792	4695	816	404	632
NSLU : number of u-barriers	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
NSLUV : number of u- + v-barriers	0	2	1	0	34	12	22	22	0	10	0	17	0	6	62	0	17	2	2	8
NSLV : number of v-barriers	0	2	1	0	34	12	22	22	0	8	0	17	0	6	62	0	17	2	2	8
NTO : total number of tide openings	23	2	2	189	145	141	53	53	178	2	5	211	2	4	6	2	211	2	2	31
IADLND : address of inactive points in LGRID	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
KURFLG	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
NROU : number of weirs	0	0	0	0	0	0	0	0	0	61217	2794	24725	0	91470	0	0	9348	0	0	0
KMAX : maximum number of layers	1	1	1	1	1	1	1	1	1	1	1	1	4	1	1	1	1	1	1	1
ISILOR : sill depth orientation flag with respect to user input	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0
IDEPOR : depth orientation flag with respect to user input.	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0
IRLFLG : flag indicating whether curv. coord.	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NBARU : number of u-barrier points	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0
NBARV : number of v-barrier points	0	2	1	0	34	12	22	22	0	52	0	17	0	24	62	0	17	2	2	8
NBARUV : number of barrier points	0	2	1	0	34	12	22	22	0	56	0	17	0	24	62	0	17	2	2	8
NTOPT : number of opening points	327	32	17	752	1160	643	351	352	814	27	12	902	10	89	209	84	861	101	50	297
NSLUI : number of u-barriers interior to a (sub-)domain	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
NSLVI : number of v-barriers interior to a (sub-)domain	0	2	1	0	34	12	22	22	0	8	0	17	0	6	62	0	17	2	2	8
NBARUI : number of u-barrier points interior to a (sub-)domain	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0
NBARVI : number of v-barrier points interior to a (sub-)domain	0	2	1	0	34	12	22	22	0	52	0	17	0	24	62	0	17	2	2	8
NSLUVG : global number of u- and v-barriers	0	2	1	0	34	12	22	22	0	10	0	17	0	6	62	0	17	2	2	8
ICOCOD : grid type code	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3.4 Parameters and properties of the simulation models

3.12 WAQUA programs

pre processor WAQPRE	The first program is the WAQUA pre-processor WAQPRE which checks the user input and prepares data for the simulation program WAQPRO. WAQPRE completes data preparation for the processor program WAQPRO, mainly by combining various time-varying inputs into strict time order for sequential processing across simulated time.
processor WAQPRO	The next program is the 2-dimensional water quality simulation program WAQPRO. The simulation program is large in terms of core storage requirements and in terms of run time.
restart with WAQPRO and WAQPRE	Because of the long run time for a simulation, the simulation program WAQPRO writes on request restart data to the SDS file. All com-putations throughout the field are saved, so that the program can be restarted later at a requested time, after rerunning WAQPRE.
post-processor WAQVIEW	For the SIMONA postprocessing program WAQVIEW refer to the user's guide WAQVIEW.
WAQWND	WAQWND converts a file delivered by the KNMI describing space varying wind and pressure into a wind SDS file.

Appendices

Appendix A

Examples

A.1 Example user routine

A.1.1 General

The aim of the following example is not to solve the problem in a correct manner, but to demonstrate the utilities of the user routine WASUST and of the SIMONA processing tool SIFODE. Also is shown how to write results to a self defined file within the SIMONA system.

A.1.2 Model description

In this example we try to model a so-called mussel filter. This filter consists of an amount of parallel nets, which are placed perpendicular to the bottom and perpendicular to the stream flow. Each net contains several mussels.

The mussels capture suspended matter which is contaminated with cadmium. A part of the cadmium is accumulated by the mussels in their tissues; the rest ends up in pseudofaeces. The pseudofaeces consist of smaller particles and sedimentate. Consequently the concentration of suspended matter in water and hence the concentra-tion of cadmium decreases. Expressions for filtering and pseudofaeces production are taken from *'Filtration rate and pseudofaeces produc-tion in Zebra Mussels and their application in water quality manage-ment' (R. Noordhuis et. al., Limnologie Aktuell (4) 1992)*.

The schematisation of the water system in this model is chosen as small as possible: M = 3 and N = 3, the filter is positioned in the middle (M = 2, N = 2). At one boundary a discharge of 18 m^3/s is imposed with a concentration of suspended matter of 12 g/m^3, which is also the initial concentration. At the other boundary a water level of 0 m is imposed; the overall depth is 5 m. This resembles the situation of Lake Volkerak.

Besides suspended matter and Cadmium in a mussel also 'continuity' is modelled to check the mass conservation of the numerical scheme.

The filter starts when the water movement is steady; this occurs after about one day (1440 minutes). The complete simulation takes two days.

The mathematical description of the filter is defined by two partial differential equations.

The total change of suspended matter is described by:

$$\frac{\delta}{\delta t}(\text{Suspended matter}) \quad \begin{array}{l} \text{- "advection"} \\ \text{+ "dispersion"} \\ \text{- "filtration by mussels"} \end{array}$$

In general only the first two terms are solved by the standard transport solver. The solver also accounts for the last term, when the user routine WASUST is called, where sources or sinks get a value defined by the user.

The last term acts as a 'sink'. This term is proportional to the concentration of suspended matter, the filtration rate of a single mussel and the total amount of mussels. The filtration rate itself depends on the concentration of suspended matter, which actually makes this term unlinear; nevertheless in this example implicit modelling (SIMP) is applied. The change of Cadmium in a single mussel is described by:

$$\frac{d}{dt}(\text{Cadmium in mussel}) \quad \begin{array}{l} \text{+ "intake"} \\ \text{- "faeces production"} \end{array}$$

The constituent 'cadmium in a mussel' is not a constituent in terms of WAQUA: it is not effected by advection or dispersion. For that reason the array SOLUSR and the routine SIFODE are available. SIFODE is solving an ordinary differential equation: for linear (sink) terms coefficient α has to be defined, all other terms use coefficient β. In fact the routine SIFODE does the same as the transport solver of WAQUA/TRIWAQ without advection and dispersion. Coefficient α can be compared with SIMP, coefficient β with SEXP. In this example both 'intake' and 'faeces production' are modelled explicitly by using coefficient beta.

Routine SIFODE has to be called by the user, unlike WASUST, which is called automatically by the system.

A more detailed description can be found in the source code (pseudo code and comments). Results are graphically shown in Fig. A.1 and Fig. A.2.

A.1.3 Input file mussel filter

```
# 2 constituents: continuity and suspended matter
# run 001 number of nets = 100 ; density mussels per m2 = 5000
# discharge = 18 m3/s; filter calculations after 1440 minutes
```

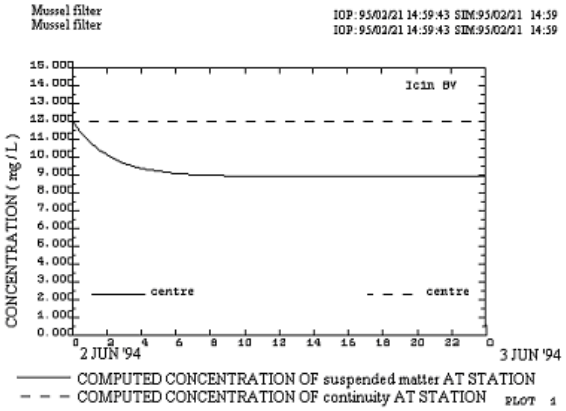


Figure A.1:

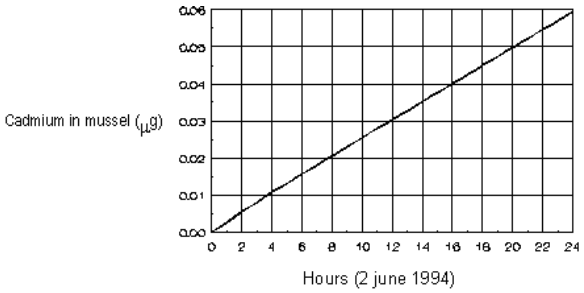


Figure A.2:

```
# Cadmium calculation

IDENTification
    WAQUA
    EXPERIMENT='001' OVERWRITE
    MODID='Cdmussel'
    TITLE='Mussel filter'

MESH
    GRID AREA
        ANGLEgrid = 0.0   # angle Y axis and North
        MMAX = 3    NMAX = 3   # number of grids in X and Y
        LATItude = 52.5

    RECTilinear STEPsize = 500.0   # distance between grid points

POINTS   # checkpoints
    P 1=(M=1,N=2,NAME='left')
    P 2=(M=2,N=2,NAME='centre')
    P 3=(M=3,N=2,NAME='right')
```

```
    BOUNdaries
    OPENings
        OPEN1: LINE (P1 , P1  NAME = 'Q opening')
        OPEN2: LINE (P3 , P3  NAME = 'W level opening')

    BATHYMetry GLOBAL CONST_VALUES =5.0

GENERAL
    DIFFusion GLOBAL CONST_values = 25   # local diffusion coef.
    PHYSicalparameters
    WATDENsity= 1000.0

FLOW
    PROBlem
    TIMEFrame DATE= '01 jun 1994'  TSTART= 0.    TSTOP= 2880.    # 2 days
    METHODvariables
        TSTEP= 1   # time step in minutes
        ITERACCURVel= 0.005   # convergence criterium flow
           # velocities

    FRICTion
        GLOBAL
        UDIRec GLOBAL CONST_values = 0.026     # friction value
        VDIRec GLOBAL CONST_values = 0.026     # friction value

    FORCings
       INITial

       BOUNDAries
           B: OPEN1 BTYPe='disch' BDEF='series' SAME
           B: OPEN2 BTYPe='wl'     BDEF='series' SAME

       TIMESERies
           S: P1 TID= 18.0     # discharge 18 m3/s
           S: P3 TID=  0.0     # water level

    CHECKPoints
        LEVELStations    P1, P2, P3
        CURRENtstations P1, P2

TRANSPORT
    PROBlem
    CONSTITuents
        CO1: POLUTant = 'continuity',     PUNIT = 'mg/l'
        CO2: POLUTant = 'suspended matter', PUNIT = 'mg/l'
```

```
    FORCings
    INITial
        CONSTITUents
        CO1  GLOBAL CONST_values = 12.0     # g/m3 = mg/l
        CO2  GLOBAL CONST_values = 12.0     # g/m3 = mg/l

    BOUNDaries
        RETURNtime
        CRET: P1, TCRETA=1    # Return time after current ...
        CRET: P3, TCRETA=1    # ... reverse to inward flow
        TIMESeries
         TS: CO1, P1, CINIT = 12.0
         TS: CO2, P1, CINIT = 12.0

    CHECKPOINTS
        CONSTITUENT_stations  P1, P2, P3

    USERData_transport
    CONTRol
        TYPE = 1 LENWrk = 0
    REALS
        USER1 = 1440.    # starting time effect mussel filter
        USER2 = 2880.    # = TSTOP (needed for closing file in WASUST)
        USER3 =   30.    # time interval writing results in WASUST

    INTEGers
        IUSER1: 5000     # density of mussels per square m
        IUSER2:  100     # amount of nets in filter
        IUSER3:    1     # identify number for output file

    OUTPUT_SPatial_data
        OS1
        NAMES    NAME = 'Cd_mussel'
                 UNIT = 'micro g/mussel'
        GLOBAL     CONST_value = 0.0     # starting values

SDSOUTput
    MAPS TFMAPs=  1440.  TIMAPs= 1440.  TLMAPs=  2880.
    HISTories TFHISto= 1440.  TIHISto=  30.  TLHISto= 2880.

PRINToutput
    TRANSport CO1 CO2
    CONTRol TFRAMEHist=1440,30,2880
```

A.1.4 Subroutine WASUST

```
      subroutine wasust(irogeo,kh,guuvv,xydep,rp,vel,wind,wval,
     + zk,difuv,difw,vicow,rho,user,iuser,
     + fvalue,spainp,spatim,solusr,work,simp,
     + sexp,dum1,dum2,dum3)
c
c=======================================================================
c   Programmers   Carlijn Bak, Dirk Vlag
c   Version 1.0   Date    21 2 1995
c   Local filename  mosfil.f
c
c
c   Copyright (c) 1993 "Rijkswaterstaat".
c   Permission to copy or distribute this software or documentation
c   in hard copy or soft copy granted only by written license
c   obtained from "Rijkswaterstaat".
c   All rights reserved. No part of this publication may be
c   reproduced, stored in a retrieval system (e.g., in memory, disk,
c   or core) or be transmitted by any means, electronic, mechanical,
c   photocopy, recording, or otherwise, without written permission
c   from the publisher.
c
c ***********************************************************************
c
c   DESCRIPTION
c
c Example user routine wasust. Empty body is filled with code to
c describe the simulation of a mussel filter: decreasing suspended
c matter by use of a mussel filter.
c Cadmium balance, suspended matter, mussel and pseudo faeces.
c Musselfilter in the centre of the WAQUA grid(M = 2, N = 2).
c Effect filter starts one day after the beginning of the experiment.
c
c ***********************************************************************
c
c      KEYWORDS
c
c   simona
c   processor
c   transport
c   user_routines
c
c ***********************************************************************
c
```

```
c       INPUT / OUTPUT    PARAMETERS
c
c Because of the length this part is shortened. A complete description
c can be found in User's guide WAQUA, section 5 (processor WAQPRO).
c
c   iuser(1)        i    mussel density on a net [number of mussels/m2]
c   iuser(2)        i    number of nets
c   iuser(3)        i    identify number output file
c   user(1)         i    starting time of effect filter (1440 min.)
c   user(2)         i    = TSTOP
c   user(3)         i    time interval printing results to own file
c   solusr(2,2,1)   o    Cadmium in a mussel [micro g/mussel]
c
c ***************************************************************
c
c    LOCAL PARAMETERS
   integer iinput(3),info(4),irefun,mvak,nvak
   real alfa(1,1),beta(1,1),c(1,1),cd_in,cd_psp,cf,cp
   real dt,fr,rinput(1),r1,r2,r3,r4,r5,totmus
   character fname*100,name*6
   parameter (mvak=2,nvak=2)
   parameter (r1=187.1,r2=0.037,r3=4.e 6,r4=5.54,r5=0.97)
   parameter (cf=1.e 6/3600.,cp=1.e 3/(24.*3600.))
   save       alfa,c,dt,iinput,irefun,totmus
   data name /'WASUST'/
c
c   alfa   'full matrix' form alfa coefficient [1/s] (SIFODE)
c   beta   'full matrix' form beta coefficient [g/s/mussel]
c         (SIFODE)
c   c      'full matrix' form with results from SIFODE [g/mussel]
c   cd_in   mussel intake of cadmium [g/s/mussel]
c   cd_psp    cadmium in pseudo faeces [g/s/mussel]
c   cf   conversion [ml/h]   > [m3/s]
c   cp   conversion [mg/day]   > [g/s]
c   dt   dtsec/2
c   fname   filename for own results
c   fr   filtration rate [m3/s/mussel] (in Noordhuis
c         [ml/h/mussel])
c   iinput    (1) length of IINPUT - 1 (SIFODE)
c         (2) flag indicating:
c            0: time integration over all grid points
c            1: IROGEO table is used
c   info   (1)  status file: 0 = old, 1 = new, 2 = old/new
c         (2)  access file: 0 = seq., 1 = direct
c         (3)  format file: 0 = formatted, 1 = unformatted
```

```
c         (4)  record length for direct
c   irefun    ref. to unitnr. of fname
c   mvak   location filter in WAQUA grid
c   name   name of subroutine (WASUST)
c   nvak   location filter in WAQUA grid
c   r1   constant {Noordhuis et. al.}
c   r2   constant {Noordhuis et. al.}
c   r3   constant (fraction of Cd in susp. matter)
c   r4   constant {Noordhuis et. al.}
c   r5   constant {Noordhuis et. al.}
c   rinput    not yet used (SIFODE)
c   totmus    total amount of mussels
c
c ***************************************************************
c
c   I / O
c
c   /export/home/bak/waqua/MOSSEL/mosout.???
C   with ??? = iuser(3)
c   The unit number of this file is supplied by the Simona system
c
c ***************************************************************
c
c   SUBROUTINES CALLED
c
c   SIFLCL   : closing file with Cd results
c   SIFLOP   : opening file with Cd results
c   SIFODE   : computing amount of Cd in one mussel after a time step
c   SITXRB   : get position last non blank character in string
c
c ***************************************************************
c
c   ERROR MESSAGES
c
c   None.
c
c ***************************************************************
c
c   PSEUDO CODE
c
c   d(SM*Vol)/dt =   fr * totmus * SM,   Vol = h*dx*dy
c    SM := rp(.,.,.,2) suspended matter [mg/l]
c    simp(.,.,.,2) = fr * totmus / Area,   Area = dx*dy
c
c   d(cd_mus)/dt = fr * SM * r3   PSP * r3
```

```
c    cd_mus := solusr(.,.,1) Cadmium in a mussel [micro g/mussel]
c    PSP := pseudo faeces production [micro g/s/mussel]
c
c   use routine SIFODE with beta = fr * rp(.,.,.,2) * r3   PSP * r3
c
c ================================================================
c
   if (((timnow tstart)*60.)   dtsec .lt.  0.1) then
c
c First call, initializing
c
c Calculating total amount of mussels
c
      totmus = iuser(1)*iuser(2) * dy * ( 1. * zk(nvak,mvak,1))
c
c Initial values needed for SIFODE
c
      dt=dtsec/2.
      alfa(1,1)=0.
      c(1,1)=solusr(nvak,mvak,1)*1.0e 6
      iinput(1)=2
      iinput(2)=0
c
c Setting up and opening the output file for recording Cd in one mussel
c
      fname='mosout.'
c
c Determine length of fname with simona tool sitxrb
c
      call sitxrb(fname,ifnb,ilnb)
      write(fname(ilnb+1:ilnb+3),'(i3.3)') iuser(3)
      info(1)=2
      info(2)=0
      info(3)=0
      call siflop(fname,info,irefun,name)
   endif
c
c End of initializing
c
   if(timnow .gt. user(1)) then
c
c Computing effect mussel nets after user(1) minutes on suspended
c matter
c
      fr = r1 * exp(  r2 * rp(nvak,mvak,1,2)) * cf
```

```
      simp(nvak,mvak,1,2) = fr * totmus / (dx*dy)
c
c Computing accumulation of Cd in one mussel
c
      cd_in = fr * rp(nvak,mvak,1,2) * r3
      cd_psp = (r4 + r5 * rp(nvak,mvak,1,2)) * cp * r3
      beta(1,1) = cd_in   cd_psp
      call sifode(c,alfa,beta,1,1,dt,iinput,rinput,irogeo,norows)
c
c Writing to solusr (in ug/mussel)
c
      solusr(nvak,mvak,1)=c(1,1)*1.0e+6
c
c Every user(3) time interval writing Cd to file
c
      if(mod(timnow,user(3)) .lt. 0.01) then
      write(ireffl(1,irefun),'(2g12.5)')
   +      timnow,solusr(nvak,mvak,1)
      endif
   else
c
c timnow < user(1): no effect mussel nets
c
      simp(nvak,mvak,1,2) = 0.
      solusr(nvak,mvak,1) = 0.
   endif
c
c At TSTOP(=user(2)) closing the file with Cd results
c
   if(abs(user(2) timnow) .lt. .001) then
      call siflcl(irefun,name)
   endif
   end
```

Appendix B

Definition WAQUA standard version

The WAQUA system as it is available for users consists of a series of subsystems. Rijkswaterstaat defines which subsystems belong to the WAQUA system. Maintenance and support is carried out by Deltares.

This so called standard version of the WAQUA system is constantly evolving. Every year the most up-to-date standard version forms the basis of the major release of Simona.

B.1 Standard version of May 2012

B.1.1 SIMONA standard version subsystems

WAQIPW:	interactively inspects and modifies a WAQUA-Input File.
WAQWND:	converts KNMI wind files to an SDS format.
WAQPRE:	processes and checks the extensive WAQUA input consisting of a structured ASCII-file.
OBS2SDS:	add observations to an SDS file.
WAQPRO:	executes two- and three-dimensional water movement, water quality and silt computations, storing calculation results on an SDS file; is able to perform rectilinear, curvilinear and spherical computations.
WAQVIEW:	interactive postprocessor; with extra river functionality.
KALGUI:	interactive postprocessor; with extra kalman functionality.
GETDATA:	conversion of SDS-file to several formats, such as: NetCDF, Matlab, shapefiles and ascii.
MODNST:	generates boundary conditions using nesting.
SDS2MAT:	conversion of SDS-file to Matlab (to be used in Kalgui)
SIMPAR:	calculates the displacement of particles in a two-dimensional water flow environment.
SLIB3D:	computes the distribution of silt.

Appendix C

Harmonic Constants

nr	Component	grad/h	rad/min	10E-4 rad/s
1	SA	.041069	.000012	.001991
2	SSA	.082137	.000024	.003982
3	MSM	.471521	.000137	.022860
4	MM	.544375	.000158	.026392
5	MSF	1.015896	.000296	.049252
6	SM	1.015896	.000296	.049252
7	MF	1.098033	.000319	.053234
8	SNU2	1.487417	.000433	.072112
9	SN	1.560270	.000454	.075644
10	MFM	1.642408	.000478	.079626
11	2SM	2.031792	.000591	.098504
12	2SMN	2.576166	.000749	.124896
13	2Q1	12.854286	.003739	.623193
14	NJ1	12.854286	.003739	.623193
15	SIGMA1	12.927140	.003760	.626725
16	NUJ1	12.927140	.003760	.626725
17	Q1	13.398661	.003898	.649585
18	RO1	13.471514	.003919	.653117
19	NUK1	13.471514	.003919	.653117
20	O1	13.943036	.004056	.675977
21	TAU1	14.025173	.004080	.679960
22	MP1	14.025173	.004080	.679960
23	M1B	14.487410	.004214	.702369
24	M1C	14.492052	.004216	.702595
25	M1D	14.492052	.004216	.702595
26	M1A	14.496694	.004217	.702820
27	M1	14.496694	.004217	.702820
28	NO1	14.496694	.004217	.702820
29	CHI1	14.569548	.004238	.706352
30	LP1	14.569548	.004238	.706352

31	PI1	14.917865	.004339	.723238
32	TK1	14.917865	.004339	.723238
33	P1	14.958931	.004351	.725229
34	S1	15.000000	.004363	.727221
35	K1	15.041069	.004375	.729212
36	PSI1	15.082135	.004387	.731203
37	RP1	15.082135	.004387	.731203
38	FI1	15.123206	.004399	.733194
39	KP1	15.123206	.004399	.733194
40	THETA1	15.512590	.004512	.752072
41	LABDAO1	15.512590	.004512	.752072
42	J1	15.585443	.004534	.755604
43	2PO1	15.974827	.004647	.774481
44	SO1	16.056965	.004671	.778464
45	OO1	16.139103	.004695	.782446
46	KQ1	16.683475	.004853	.808838
47	3MKS2	26.870174	.007816	1.302703
48	3MS2	26.952312	.007840	1.306685
49	OQ2	27.341696	.007953	1.325563
50	MNK2	27.341696	.007953	1.325563
51	MNS2	27.423834	.007977	1.329545
52	2ML2S2	27.496687	.007998	1.333077
53	2MS2K2	27.803934	.008088	1.347973
54	NLK2	27.886070	.008112	1.351955
55	2N2	27.895355	.008114	1.352405
56	MU2	27.968208	.008136	1.355937
57	2MS2	27.968208	.008136	1.355937
58	SNK2	28.357592	.008249	1.374815
59	N2	28.439730	.008273	1.378797
60	NU2	28.512583	.008294	1.382329
61	2KN2S2	28.604004	.008321	1.386761
62	OP2	28.901966	.008407	1.401207
63	MSK2	28.901966	.008407	1.401207
64	MPS2	28.943035	.008419	1.403198
65	M2	28.984104	.008431	1.405189
66	MSP2	29.025173	.008443	1.407180
67	MKS2	29.066240	.008455	1.409171
68	M2(KS)2	29.148378	.008479	1.413153
69	2SN(MK)2	29.373487	.008544	1.424067
70	LABDA2	29.455626	.008568	1.428049
71	2MN2	29.528479	.008589	1.431581
72	L2	29.528479	.008589	1.431581
73	L2A	29.528479	.008589	1.431581
74	L2B	29.537764	.008592	1.432031
75	2SK2	29.917864	.008703	1.450459

76	T2	29.958933	.008715	1.452450
77	S2	30.000000	.008727	1.454441
78	R2	30.041067	.008739	1.456432
79	K2	30.082136	.008751	1.458423
80	MSN2	30.544374	.008885	1.480833
81	ETA2	30.626513	.008909	1.484815
82	KJ2	30.626513	.008909	1.484815
83	MKN2	30.626513	.008909	1.484815
84	2KM(SN)2	30.708649	.008933	1.488797
85	2SM2	31.015896	.009022	1.503693
86	SKM2	31.098034	.009046	1.507675
87	2SNU2	31.487417	.009159	1.526553
88	3(SM)N2	31.487417	.009159	1.526553
89	2SN2	31.560270	.009181	1.530085
90	SKN2	31.642408	.009204	1.534067
91	MQ3	42.382767	.012329	2.054775
92	NO3	42.382767	.012329	2.054775
93	MO3	42.927139	.012487	2.081166
94	2MK3	42.927139	.012487	2.081166
95	2MP3	43.009277	.012511	2.085149
96	M3	43.476154	.012647	2.107783
97	SO3	43.943035	.012783	2.130418
98	MK3	44.025173	.012806	2.134401
99	2MQ3	44.569550	.012965	2.160793
100	SP3	44.958931	.013078	2.179670
101	SK3	45.041069	.013102	2.183653
102	K3	45.123207	.013126	2.187635
103	2SO3	46.056965	.013397	2.232905
104	4MS4	55.936417	.016271	2.711874
105	2MNS4	56.407936	.016408	2.734734
106	3MK4	56.870174	.016543	2.757144
107	MNLK4	56.870174	.016543	2.757144
108	3MS4	56.952312	.016567	2.761126
109	MSNK4	57.341698	.016680	2.780004
110	MN4	57.423836	.016704	2.783986
111	2MLS4	57.496689	.016725	2.787518
112	2MSK4	57.886070	.016838	2.806396
113	M4	57.968208	.016862	2.810378
114	2MKS4	58.050346	.016886	2.814360
115	SN4	58.439732	.016999	2.833238
116	3MN4	58.512585	.017021	2.836770
117	2SMK4	58.901966	.017134	2.855648
118	MS4	58.984104	.017158	2.859630
119	MK4	59.066242	.017182	2.863612
120	2SNM4	59.455624	.017295	2.882490

121	2MSN4	59.528481	.017316	2.886022
122	SL4	59.528481	.017316	2.886022
123	S4	60.000000	.017453	2.908882
124	SK4	60.082138	.017477	2.912864
125	2SMN4	60.544376	.017612	2.935274
126	3SM4	61.015896	.017749	2.958134
127	2SKM4	61.098034	.017773	2.962116
128	MNO5	71.366867	.020760	3.459963
129	3MK5	71.911247	.020918	3.486356
130	3MP5	71.993378	.020942	3.490337
131	M5	72.464905	.021079	3.513198
132	MNK5	72.464905	.021079	3.513198
133	2MP5	72.927139	.021214	3.535607
134	3MO5	73.009277	.021238	3.539590
135	MSK5	74.025169	.021533	3.588841
136	3KM5	74.107307	.021557	3.592824
137	2(MN)S6	84.847664	.024681	4.113531
138	3MNS6	85.392044	.024840	4.139923
139	2NM6	85.863564	.024977	4.162783
140	4MS6	85.936417	.024998	4.166315
141	2MSNK6	86.325798	.025111	4.185193
142	2MN6	86.407936	.025135	4.189175
143	2MNU6	86.480789	.025156	4.192707
144	3MSK6	86.870178	.025270	4.211585
145	M6	86.952316	.025293	4.215567
146	MSN6	87.423836	.025431	4.238427
147	4MN6	87.496689	.025452	4.241959
148	MNK6	87.505974	.025454	4.242409
149	MKNU6	87.578827	.025476	4.245941
150	2(MS)K6	87.886070	.025565	4.260837
151	2MS6	87.968208	.025589	4.264819
152	2MK6	88.050346	.025613	4.268801
153	2SN6	88.439728	.025726	4.287679
154	3MSN6	88.512581	.025747	4.291211
155	MKL6	88.594719	.025771	4.295193
156	2SM6	88.984100	.025884	4.314071
157	MSK6	89.066238	.025908	4.318053
158	S6	90.000000	.026180	4.363323
159	2MNO7	100.350975	.029191	4.865153
160	2NMK7	100.904633	.029352	4.891995
161	M7	101.449005	.029510	4.918387
162	2MSO7	101.911247	.029645	4.940797
163	MSKO7	103.009277	.029964	4.994031
164	2(MN)8	114.847664	.033408	5.567972
165	3MN8	115.392044	.033566	5.594364

166	3MNKS8	115.474182	.033590	5.598346
167	M8	115.936417	.033725	5.620756
168	2MSN8	116.407936	.033862	5.643616
169	2MNK8	116.490074	.033886	5.647598
170	3MS8	116.952316	.034020	5.670008
171	3MK8	117.034447	.034044	5.673990
172	2SNM8	117.423836	.034157	5.692868
173	MSNK8	117.505974	.034181	5.696850
174	2(MS)8	117.968208	.034316	5.719260
175	2MSK8	118.050346	.034339	5.723242
176	3SM8	118.984100	.034611	5.768512
177	2SMK8	119.066238	.034635	5.772494
178	S8	120.000000	.034907	5.817764
179	2(MN)K9	129.888733	.037783	6.297183
180	3MNK9	130.433105	.037941	6.323575
181	4MK9	130.977493	.038100	6.349968
182	3MSK9	131.993378	.038395	6.399220
183	4MN10	144.376144	.041997	6.999553
184	M10	144.920517	.042156	7.025945
185	3MSN10	145.392044	.042293	7.048805
186	4MS10	145.936417	.042451	7.075197
187	2(MS)N10	146.407944	.042588	7.098057
188	2MNSK10	146.490082	.042612	7.102040
189	3M2S10	146.952316	.042747	7.124449
190	4MSK11	160.977493	.046826	7.804409
191	M12	173.904633	.050587	8.431135
192	4MSN12	174.376144	.050724	8.453994
193	5MS12	174.920517	.050882	8.480386
194	3MNKS12	175.474182	.051043	8.507228
195	4M2S12	175.936417	.051178	8.529638